

THE USE OF MACHINE LEARNING ALGORITHMS TO DERIVE FRAGILITY
CURVES FOR MID-RISE REINFORCED CONCRETE BUILDINGS

by

Onur Ülkü

B.S., Civil Engineering, Yıldız Technical University, 2019

Submitted to the Kandilli Observatory and Earthquake
Research Institute in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Earthquake Engineering
Boğaziçi University

2023

ACKNOWLEDGEMENTS

I would like to convey my sincere appreciation to Assoc. Prof. Ufuk Hancilar for his exceptional guidance, constant support, and essential mentorship throughout the process of my thesis. His insightful comments, helpful criticism, and support have continuously inspired me to pursue my goals, allowing me to advance academically and personally.

I also would like to extend my heartfelt gratitude to my colleague, Ali Talha Atici, for his invaluable collaboration, dedication, countless brainstorming sessions, late-night discussions, and the exchange of ideas that we have shared.

Moreover, I would like to thank my co-workers, Elif Yıldırım and Nurullah Acikgoz, for their collaboration, helpful suggestions, and feedback.

Last but not least, I have always been thankful to my father, my mother, and my girlfriend, Aybala Demir, for their immense support and encouragement.

ABSTRACT

THE USE OF MACHINE LEARNING ALGORITHMS TO DERIVE FRAGILITY CURVES FOR MID-RISE REINFORCED CONCRETE BUILDINGS

The occurrence of an earthquake does not necessarily indicate that there is a seismic risk. The existence of risk depends on having three components, which are hazards, exposures, and fragility together. Assessing the risk of existing buildings is a building-specific task that may be tremendously time-consuming and computationally burdensome. Moreover, determining the risk of each structure can be complicated when investigating a portfolio or a group of buildings. Developing fragility curves for buildings provides a possible and undemanding method to estimate damage likelihood. Machine learning algorithms are one of the novel approaches that are implemented for estimating potential structural damage. Well-trained machine learning algorithms provide to speed up processing, cut down the cost of computation, and produce reliable fragility curves. This thesis focuses on developing fragility curves for generic building inventory spread over the Marmara region by predicting the probability of maximum inter-story drift ratio intervals via five various machine learning algorithms, which are Random Forest, Stochastic Gradient Boosting, Naïve Bayes, Decision Tree, K-Nearest Neighbors. Information on the characteristics of pre-dominant building typologies in the Marmara region available in the literature is utilized for creating an artificial inventory dataset of mid-rise RC buildings. The data for the machine learning was gathered by designing and analyzing 2-D frame systems under non-linear time history analysis with OpenSeesPy. The machine learning algorithms are trained considering different intensity measures and buildings' characteristics. Machine learning algorithms are evaluated when generating fragility functions by comparing those produced by fitting log-normal distribution with the maximum likelihood estimation method.

ÖZET

ORTA KATLI BETONARME BİNALAR İÇİN KIRILGANLIK EĞRİLERİNİN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TÜRETİLMESİ

Bir depremin meydana gelmesi, mutlaka bir sismik risk olduğunu göstermez. Riskin varlığı; bir konumda deprem tehlikesinin var olması, o deprem tehlikesine maruz kalacak bir varlığın olması ve o varlığın deprem tehlikesi altında hasar görebilme durumunu içermesine bağlıdır. Mevcut binaların riskini değerlendirmek, son derece zaman alıcı ve binaya özgü bir görevdir. Bununla birlikte, bir portföy veya bir grup bina incelenirken her bir yapının riskini belirlemek karmaşık olabilir. Binalar için kırılma eğrileri geliştirmek, potansiyel hasarı belirlemek için olası ve daha anlaşılabilir bir yöntem sağlar. Makine öğrenimi algoritmaları, potansiyel yapısal hasarı tahmin etmek için yeni yaklaşımlardan biridir. İyi eğitilmiş makine öğrenimi algoritmaları, işlemlerin hızlandırılmasını, hesaplama maliyetinin düşürülmesini ve güvenilir kırılma eğrileri üretilmesini sağlar. Bu tez, Random Forest, Stokastik Gradient Boosting, Naïve Bayes, Decision Tree, K-Nearest Neighbor olmak üzere beş farklı makine öğrenimi algoritması ile maksimum katlar arası öteleme oranı aralıklarının olasılığını tahmin ederek Marmara bölgesinde bulunan jenerik bina envanteri için kırılma eğrileri geliştirmeye odaklanmaktadır. Yapay olarak oluşturulan orta katlı betonarme binaların yapısal özellikleri, Marmara bölgesi için yapılmış literatür araştırmaları dikkate alınarak belirlenmiştir. Makine öğrenmesi için veriler, OpenSeesPy paketi ile 2 boyutlu çerçeve sistemleri tasarlanarak ve bu tasarlanan modeller zaman tanım alanında doğrusal olmayan analiz edilerek toplanmıştır. Makine öğrenimi algoritmaları, farklı yer hareketi şiddet parametreleri ve binaların özellikleri dikkate alınarak eğitilmiştir. Makine öğrenmesi algoritmaları ile üretilen kırılma fonksiyonları maksimum olabilirlik tahmini ile log-normal dağılıma oturtturulmuş kırılma fonksiyonları ile karşılaştırılarak değerlendirilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	viii
LIST OF TABLES	xxii
LIST OF SYMBOLS	xxviii
LIST OF ACRONYMS/ABBREVIATIONS	xxix
1. INTRODUCTION	1
1.1. Objectives and Outline of the Thesis	3
1.2. Literature Review	4
2. PREPARATION OF DATASET AND MODEL OF STRUCTURES	8
2.1. Introduction	8
2.2. Building Characteristics of the Stock in the Marmara Region	10
2.2.1. Physical Properties of Representative RC Buildings	12
2.2.2. Material Properties of Representative RC Buildings	17
2.3. Finite Element Models of the Representative Buildings	19
2.3.1. Verification of the Models	25
2.4. Assessment of the Structural Response of the Models	25
2.4.1. Selection of Ground Motion Records	28
2.4.2. Intensity Measures of Ground Motion	33
2.4.3. Engineering Demand Parameters	35
2.4.4. Evaluation of Dynamic Structural Analysis Results	35
3. APPLICATION OF MACHINE LEARNING ALGORITHMS	40
3.1. Introduction	40
3.2. Machine Learning Algorithms	42
3.2.1. Decision Tree	43
3.2.2. Random Forest	44
3.2.3. K-Nearest Neighbor	44

3.2.4. Naïve Bayes	45
3.2.5. Stochastic Gradient Boosting	46
3.3. Metrics of Evaluating the Model Performance	47
3.3.1. Accuracy	47
3.3.2. Confusion matrix	48
3.3.3. Precision and Recall	48
3.3.4. F_1 Score	49
3.4. Data Processing for Machine Learning Algorithms	49
3.4.1. Raw Data	49
3.4.2. Pre-Processing of Raw Data	63
4. THE RESULT OF MACHINE LEARNING ALGORITHMS AND THE EVAL- UATION OF PERFORMANCE METRICS OF THE ALGORITHMS	76
4.1. Derivation of Fragility Curves via ML Algorithms	84
4.2. Derivation of Fragility Curves Through Log-normal Distribution	100
4.3. Comparison of Fragility Curves Derived from ML and Log-normal Dis- tribution	105
4.4. Comparison of Fragility Curves Derived in this Thesis with the Ones Used in Istanbul Loss Assessment Studies	121
5. CONCLUSION	126
5.1. Summary of Findings	128
5.2. Recommendation for Further Studies	130
REFERENCES	132
APPENDIX A: SCRIPTS OF GENERATING MODEL, NONLINEAR TIME HISTORY ANALYSIS, DATA PREPARATION, MACHINE LEARNING ALGORITHMS, AND DERIVING FRAGILITY CURVES	140
APPENDIX B: THE SELECTED GROUND MOTIONS	141
APPENDIX C: CONFUSION MATRICES FOR ALL ML ALGORITHMS AND ALL INTENSITY MEASURES	172

LIST OF FIGURES

Figure 2.1.	Variation of the percentage of the stock related to the construction year and number of stories of reinforced concrete frame buildings in Istanbul (DEZIM, 2020).	10
Figure 2.2.	The distribution of standard story height for all RC buildings (Bal et al., 2008).	13
Figure 2.3.	The box plot of standard and ground story height for all RC buildings (Azak et al., 2014).	13
Figure 2.4.	The distribution of span length for all RC buildings (Bal et al., 2008).	14
Figure 2.5.	The floor plan directions for the short (left panel) and long (right panel) have their mean values and matching standard deviations. (Azak et al., 2014).	15
Figure 2.6.	Statistics for average column widths of the floor plan according to story number in the short (left panel) and long (right panel) directions (Azak et al., 2014).	16
Figure 2.7.	Statistics for average column depth according to story number in the floor plan's short (left panel) and long (right panel) directions (Azak et al., 2014).	16
Figure 2.8.	Distribution of characteristic 28 th -day compressive strength of the concrete sample taken from the Marmara region (Bal et al., 2008).	18

Figure 2.9.	Yield strength distribution of S220 reinforcing steel (Bal et al., 2008 and Akyuz et al., 1992).	18
Figure 2.10.	The strain-stress models for unconfined and confined concrete under monotonic loading (Mander et al., 1988).	20
Figure 2.11.	Stress-strain model without isotropic hardening (The OpenSeesPy Framework).	21
Figure 2.12.	P-Delta effect: (a) undeformed configuration, (b) deformed configuration, (c) bending moment distribution (Burgos and Silva, 2023).	22
Figure 2.13.	The plan of the 4-span building and tributary loads for beams.	23
Figure 2.14.	Examples of models generated via OpenSeesPy for 6- and 8-story representative buildings.	24
Figure 2.15.	Modal damping ratio with respect to natural frequencies for an analyzed sampled model.	27
Figure 2.16.	Earthquake record obtained from the station 3142 after 02.06.2023 Pazarcik-Kahramanmaras Earthquake.	32
Figure 2.17.	Response spectra of selected ground motions for period class 1 and 1130 m/s V_{s30} (left) AFAD database (right) CyberShake database with median conditional mean spectrum (blue solid line) of period class 1 [Figures were plotted by using Baker and Lee's (2018) MATLAB Algorithm].	33
Figure 2.18.	The distribution of the first natural vibration period of the representative buildings.	36

Figure 2.19.	The distribution of MIDR of analysis using SF_1	37
Figure 2.20.	The distribution of MIDR of analysis using SF_2	38
Figure 2.21.	The distribution of MIDR of analysis using SF_3	38
Figure 2.22.	NLTH analysis results as MIDR for different scale factors.	39
Figure 3.1.	Two layers of the fitted best decision tree algorithm.	44
Figure 3.2.	The result of an example of the KNN classification algorithm (Pedregosa et al., 2011).	45
Figure 3.3.	The conditional cumulative distribution function of MIDR conditioned on column depth.	51
Figure 3.4.	The conditional probability density function of MIDR conditioned on column depth for 6 and 8-story.	51
Figure 3.5.	The conditional probability density function of MIDR conditioned on story height.	52
Figure 3.6.	The conditional cumulative distribution function of MIDR conditioned on story height.	52
Figure 3.7.	The conditional probability density function of MIDR conditioned on span length.	53
Figure 3.8.	The conditional cumulative distribution function of MIDR conditioned on span length.	54

Figure 3.9.	The conditional probability density function of MIDR conditioned on concrete strength.	54
Figure 3.10.	The conditional cumulative distribution function of MIDR conditioned on concrete strength.	55
Figure 3.11.	The conditional probability density function of MIDR conditioned on commercial use of the first story.	55
Figure 3.12.	The conditional cumulative distribution function of MIDR conditioned on commercial use of the first story.	56
Figure 3.13.	The conditional probability density function of MIDR conditioned on soil conditions.	56
Figure 3.14.	The conditional probability density function of MIDR conditioned on soil conditions.	57
Figure 3.15.	The conditional probability density function of MIDR conditioned on period class.	57
Figure 3.16.	The conditional cumulative distribution function of MIDR conditioned on period class.	58
Figure 3.17.	Scatter plot of MIDR vs. PGA for the representative buildings with the marginal distributions of MIDR and PGA.	59
Figure 3.18.	40. Scatter plot of MIDR vs. PGV for the representative buildings with the marginal distributions of MIDR and PGV.	59

Figure 3.19. Scatter plot of MIDR vs. PGD for the representative buildings with the marginal distributions of MIDR and PGD.	60
Figure 3.20. Scatter plot of MIDR vs. $S_a(T_1)$ for the representative buildings with the marginal distributions of MIDR and $S_a(T_1)$	60
Figure 3.21. Scatter plot of MIDR vs. $S_d(T_1)$ for the representative buildings with the marginal distributions of MIDR and $S_d(T_1)$	61
Figure 3.22. Scatter plot of MIDR vs. CAV for the representative buildings with the marginal distributions of MIDR and CAV.	61
Figure 3.23. Scatter plot of each IM against the others.	62
Figure 3.24. Boxplot to detect the outliers of IMs. [25-75 percentiles of the data (blue box), interquartile range (the outermost lines), the median values of the data (the lines between these percentiles), the outliers (the rhombus shapes)]	64
Figure 3.25. Boxplot (1.5 IQR) after deleting outliers from the data.[25-75 percentiles of the data (blue box), interquartile range (the outermost lines), the median values of the data (the lines between these percentiles), the outliers (the rhombus shapes)]	65
Figure 3.26. The probability density function of story height. (left) Actual distribution of story height, (right) standardized distribution of story height	68
Figure 3.27. Probability density function of span length. (left) Actual distribution of span length, (right) standardized distribution of span length	69

Figure 3.28.	The probability density function of concrete compression strength. (left) Actual distribution of concrete compression strength, (right) standardized distribution of concrete compression strength	69
Figure 3.29.	The probability density function of PGA (g). (left) Actual distribution of PGA, (right) standardized distribution of PGA	70
Figure 3.30.	The probability density function of PGV (cm/s). (left) Actual distribution of PGV, (right) standardized distribution of PGV . .	71
Figure 3.31.	The probability density function of PGD (cm). (left) Actual distribution of PGD, (right) standardized distribution of PGD	71
Figure 3.32.	Probability density function of $S_a(T_1)$ (g). (left) Actual distribution of $S_a(T_1)$, (right) standardized distribution of $S_a(T_1)$	72
Figure 3.33.	Probability density function of $S_d(T_1)$ (g). (left) Actual distribution of $S_d(T_1)$, (right) standardized distribution of $S_d(T_1)$	72
Figure 3.34.	The probability density function of CAV (m/s). (left) Actual distribution of CAV, (right) standardized distribution of CAV	73
Figure 4.1.	The probability mass function of MIDR levels for a random data point, dataset with PGA, obtained from the Decision Tree algorithm.	82
Figure 4.2.	The probability mass function of MIDR levels for a random data point, dataset with PGA, obtained from the Random Forest algorithm.	82
Figure 4.3.	The probability mass function of MIDR levels for a random data point, dataset with PGA obtained from the KNN algorithm. . . .	83

Figure 4.4.	The probability mass function of MIDR levels for a random data point, dataset with PGA obtained from the Naïve Bayes algorithm.	83
Figure 4.5.	The probability mass function of MIDR levels for a random data point, dataset with PGA obtained from the Stochastic Gradient Boosting algorithm.	84
Figure 4.6.	Fragility curves derived from the Decision Tree algorithm for PGA.	85
Figure 4.7.	Fragility curves derived from the Random Forest algorithm for PGA.	86
Figure 4.8.	Fragility curves derived from the KNN algorithm for PGA.	86
Figure 4.9.	Fragility curves derived from the Naïve Bayes algorithm for PGA.	87
Figure 4.10.	Fragility curves derived from the Stochastic Gradient Boosting algorithm for PGA.	87
Figure 4.11.	Fragility curves derived from the Decision Tree algorithm for PGV.	88
Figure 4.12.	Fragility curves derived from the Random Forest algorithm for PGV.	88
Figure 4.13.	Fragility curves derived from the KNN algorithm for PGV.	89
Figure 4.14.	Fragility curves derived from the Naïve Bayes algorithm for PGV.	89
Figure 4.15.	Fragility curves derived from the Stochastic Gradient Boosting algorithm for PGV.	90
Figure 4.16.	Fragility curves derived from the Decision Tree algorithm for PGD.	90

Figure 4.17. Fragility curves derived from the Random Forest algorithm for PGD.	91
Figure 4.18. Fragility curves derived from the KNN algorithm for PGD.	91
Figure 4.19. Fragility curves derived from the Naïve Bayes algorithm for PGD.	92
Figure 4.20. Fragility curves derived from the Stochastic Gradient Boosting algorithm for PGD.	92
Figure 4.21. Fragility curves derived from the Decision Tree algorithm for $S_a(T_1)$.	93
Figure 4.22. Fragility curves derived from the Random Forest algorithm for $S_a(T_1)$.	93
Figure 4.23. Fragility curves derived from the KNN algorithm for $S_a(T_1)$	94
Figure 4.24. Fragility curves derived from the Naïve Bayes algorithm for $S_a(T_1)$.	94
Figure 4.25. Fragility curves derived from the Stochastic Gradient Boosting algorithm for $S_a(T_1)$	95
Figure 4.26. Fragility curves derived from the Decision Tree algorithm for $S_d(T_1)$.	95
Figure 4.27. Fragility curves derived from the Random Forest algorithm for $S_d(T_1)$.	96
Figure 4.28. Fragility curves derived from the KNN algorithm for $S_d(T_1)$	96
Figure 4.29. Fragility curves derived from the Naïve Bayes algorithm for $S_d(T_1)$.	97
Figure 4.30. Fragility curves derived from the Stochastic Gradient Boosting algorithm for $S_d(T_1)$	97

Figure 4.31. Fragility curves derived from the Decision Tree algorithm for CAV.	98
Figure 4.32. Fragility curves derived from the Random Forest algorithm for CAV.	98
Figure 4.33. Fragility curves derived from the KNN algorithm for CAV.	99
Figure 4.34. Fragility curves derived from the Naïve Bayes algorithm for CAV.	99
Figure 4.35. Fragility curves derived from the Stochastic Gradient Boosting algorithm for CAV.	100
Figure 4.36. Fragility curves derived through log-normal distribution for PGA.	102
Figure 4.37. Fragility curves derived through log-normal distribution for PGV.	103
Figure 4.38. Fragility curves derived through log-normal distribution for PGD.	103
Figure 4.39. Fragility curves derived through log-normal distribution for $S_a(T_1)$.	104
Figure 4.40. Fragility curves derived through log-normal distribution for $S_d(T_1)$.	104
Figure 4.41. Fragility curves derived through log-normal distribution for CAV.	105
Figure 4.42. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA. . . .	106
Figure 4.43. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA. . . .	106

- Figure 4.44. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA. 107
- Figure 4.45. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA. 107
- Figure 4.46. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA. 108
- Figure 4.47. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV. 108
- Figure 4.48. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV. 109
- Figure 4.49. Comparison of the fragility curves derived from the K-Nearest Neighbor Algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV. 109
- Figure 4.50. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV. 110

Figure 4.51.	Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV.	110
Figure 4.52.	Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD. . . .	111
Figure 4.53.	Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD. . . .	111
Figure 4.54.	Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD. . . .	112
Figure 4.55.	Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.	112
Figure 4.56.	Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.	113
Figure 4.57.	Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$	113

- Figure 4.58. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$. . . 114
- Figure 4.59. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$. . 114
- Figure 4.60. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$ 115
- Figure 4.61. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$ 115
- Figure 4.62. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$. . . 116
- Figure 4.63. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$. . . 116
- Figure 4.64. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$. . 117

- Figure 4.65. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$ 117
- Figure 4.66. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$ 118
- Figure 4.67. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV. . . . 118
- Figure 4.68. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV. . . . 119
- Figure 4.69. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV. . . 119
- Figure 4.70. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV. 120
- Figure 4.71. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV. . . . 120

- Figure 4.72. Comparison of the fragility curves with respect to spectral displacement derived from the Decision Tree suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines). 122
- Figure 4.73. Comparison of the fragility curves with respect to spectral displacement derived from the Random Forest suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines). 123
- Figure 4.74. Comparison of the fragility curves with respect to spectral displacement derived from the K-Nearest Neighbor suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines). 123
- Figure 4.75. Comparison of the fragility curves with respect to spectral displacement derived from the Naïve Bayes suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines). 124
- Figure 4.76. Comparison of the fragility curves with respect to spectral displacement derived from the Stochastic Gradient Boosting suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines). 124

LIST OF TABLES

Table 2.1.	Characteristics of the representative buildings, references, and the variations of the characteristics.	11
Table 2.2.	Values of the yield strength, strain hardening ratio, and modulus of elasticity for reinforcing steel.	20
Table 2.3.	The comparison of OpenSeesPy and SAP2000 Models.	25
Table 2.4.	List of period classes and the number of analyses.	29
Table 2.5.	Example of selected ground motions with scale factors and time domain parameters for period class 1 and V_{s30} 1130 m/s.	31
Table 3.1.	Selected different data from the raw dataset.	41
Table 3.2.	Confusion matrix.	48
Table 3.3.	Features and data types of raw data.	50
Table 3.4.	Target (output) of the dataset.	50
Table 3.5.	Detection of Outliers of IMs.	63
Table 3.6.	Example of one-hot encoding.	74
Table 3.7.	An example of the final version of the dataset.	75

Table 4.1.	Evaluation metrics for the test data of machine learning algorithms are given for the specified IM.	79
Table 4.2.	Confusion matrix of RF algorithm with PGA.	80
Table 4.3.	MIDR thresholds for damage states of Mid-rise reinforced concrete buildings constructed after 1980 (Erdik et. al., 2003).	121
Table B.1.	Example of selected ground motions with scale factors and time domain parameters for period class 1 and Vs30 1130 m/s.	142
Table B.2.	Example of selected ground motions with scale factors and time domain parameters for period class 1 and Vs30 560 m/s.	143
Table B.3.	Example of selected ground motions with scale factors and time domain parameters for period class 1 and Vs30 270 m/s.	144
Table B.4.	Example of selected ground motions with scale factors and time domain parameters for period class 2 and Vs30 1130 m/s.	145
Table B.5.	Example of selected ground motions with scale factors and time domain parameters for period class 2 and Vs30 560 m/s.	146
Table B.6.	Example of selected ground motions with scale factors and time domain parameters for period class 2 and Vs30 270 m/s.	147
Table B.7.	Example of selected ground motions with scale factors and time domain parameters for period class 3 and Vs30 1130 m/s.	148
Table B.8.	Example of selected ground motions with scale factors and time domain parameters for period class 3 and Vs30 560 m/s.	149

Table B.9.	Example of selected ground motions with scale factors and time domain parameters for period class 3 and Vs30 270 m/s.	150
Table B.10.	Example of selected ground motions with scale factors and time domain parameters for period class 4 and Vs30 1130 m/s.	151
Table B.11.	Example of selected ground motions with scale factors and time domain parameters for period class 4 and Vs30 560 m/s.	152
Table B.12.	Example of selected ground motions with scale factors and time domain parameters for period class 4 and Vs30 270 m/s.	153
Table B.13.	Example of selected ground motions with scale factors and time domain parameters for period class 5 and Vs30 1130 m/s.	154
Table B.14.	Example of selected ground motions with scale factors and time domain parameters for period class 5 and Vs30 560 m/s.	155
Table B.15.	Example of selected ground motions with scale factors and time domain parameters for period class 5 and Vs30 270 m/s.	156
Table B.16.	Example of selected ground motions with scale factors and time domain parameters for period class 6 and Vs30 1130 m/s.	157
Table B.17.	Example of selected ground motions with scale factors and time domain parameters for period class 6 and Vs30 560 m/s.	158
Table B.18.	Example of selected ground motions with scale factors and time domain parameters for period class 6 and Vs30 270 m/s.	159

Table B.19.	Example of selected ground motions with scale factors and time domain parameters for period class 7 and Vs30 1130 m/s.	160
Table B.20.	Example of selected ground motions with scale factors and time domain parameters for period class 7 and Vs30 560 m/s.	161
Table B.21.	Example of selected ground motions with scale factors and time domain parameters for period class 7 and Vs30 270 m/s.	162
Table B.22.	Example of selected ground motions with scale factors and time domain parameters for period class 8 and Vs30 1130 m/s.	163
Table B.23.	Example of selected ground motions with scale factors and time domain parameters for period class 8 and Vs30 560 m/s.	164
Table B.24.	Example of selected ground motions with scale factors and time domain parameters for period class 8 and Vs30 270 m/s.	165
Table B.25.	Example of selected ground motions with scale factors and time domain parameters for period class 9 and Vs30 1130 m/s.	166
Table B.26.	Example of selected ground motions with scale factors and time domain parameters for period class 9 and Vs30 560 m/s.	167
Table B.27.	Example of selected ground motions with scale factors and time domain parameters for period class 9 and Vs30 270 m/s.	168
Table B.28.	Example of selected ground motions with scale factors and time domain parameters for period class 10 and Vs30 1130 m/s.	169

Table B.29.	Example of selected ground motions with scale factors and time domain parameters for period class 10 and Vs30 560 m/s.	170
Table B.30.	Example of selected ground motions with scale factors and time domain parameters for period class 10 and Vs30 270 m/s.	171
Table C.1.	Confusion matrix of the DT algorithm with PGA.	173
Table C.2.	Confusion matrix of the DT algorithm with PGV.	174
Table C.3.	Confusion matrix of the DT algorithm with PGD.	175
Table C.4.	Confusion matrix of the DT algorithm with $S_a(T_1)$	176
Table C.5.	Confusion matrix of the DT algorithm with $S_d(T_1)$	177
Table C.6.	Confusion matrix of the DT algorithm with CAV.	178
Table C.7.	Confusion matrix of the RF algorithm with PGV.	179
Table C.8.	Confusion matrix of the RF algorithm with PGD.	180
Table C.9.	Confusion matrix of the RF algorithm with $S_a(T_1)$	181
Table C.10.	Confusion matrix of the RF algorithm with $S_d(T_1)$	182
Table C.11.	Confusion matrix of the RF algorithm with CAV.	183
Table C.12.	Confusion matrix of the KNN algorithm with PGA.	184
Table C.13.	Confusion matrix of the KNN algorithm with PGV.	185

Table C.14.	Confusion matrix of the KNN algorithm with PGD.	186
Table C.15.	Confusion matrix of the KNN algorithm with $S_a(T_1)$	187
Table C.16.	Confusion matrix of the KNN algorithm with $S_d(T_1)$	188
Table C.17.	Confusion matrix of the KNN algorithm with CAV.	189
Table C.18.	Confusion matrix of the NB algorithm with PGA.	190
Table C.19.	Confusion matrix of the NB algorithm with PGV.	191
Table C.20.	Confusion matrix of the NB algorithm with PGD.	192
Table C.21.	Confusion matrix of the NB algorithm with $S_a(T_1)$	193
Table C.22.	Confusion matrix of the NB algorithm with $S_d(T_1)$	194
Table C.23.	Confusion matrix of the NB algorithm with CAV.	195
Table C.24.	Confusion matrix of the SGB algorithm with PGA.	196
Table C.25.	Confusion matrix of the SGB algorithm with PGV.	197
Table C.26.	Confusion matrix of the SGB algorithm with PGD.	198
Table C.27.	Confusion matrix of the SGB algorithm with $S_a(T_1)$	199
Table C.28.	Confusion matrix of the SGB algorithm with $S_d(T_1)$	200
Table C.29.	Confusion matrix of the SGB algorithm with CAV.	201

LIST OF SYMBOLS

b	Strain-hardening ratio
E_0	Elasticity modulus of the reinforcing steel
E_c	Elasticity modulus of concrete
f_c	28 th -day concrete compressive strength
F_y	Yield strength of steel
G	Dead load
Q	Live load
α	Standard Deviation
ζ	Modal damping ratio
ζ_n	nth mode modal damping ratio
μ	Median
ω_0	Natural frequency
ω_1	1 st mode natural frequency of the building
ω_2	2 nd mode natural frequency of the building

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AI	Arias Intensity
CA	Cloud Analysis
CAV	Cumulative Absolute Velocity
CoV	Coefficient of Variation
DL	Deep Learning
DT	Decision Tree
EDP	Engineering Demand Parameter
FNN	Feedforward Neural Networks
GCIM	Generalized Conditional Intensity Measure
GMM	Ground Motion Model
GMIM	Ground Motion Intensity Measure
IDA	Incremental Dynamic Analysis
IM	Intensity Measure
KNN	K-Nearest Neighbor
LBFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
MFA	Maximum Floor Acceleration
MFD	Maximum Floor Displacement
MIDR	Maximum Inter-story Drift Ratio
ML	Machine Learning
MPR	Maximum Plastic Rotation
MSA	Multiple Stripe Analysis
NB	Naïve Bayes
NLTH	Non-Linear Time History
PGA	Peak Ground Acceleration
PGD	Peak Ground Displacement
PGV	Peak Ground Velocity

RF	Random Forest
R_{jb}	Joyner-Boore Distance
S_a	Spectral Acceleration
S_d	Spectral Displacement
S_v	Spectral Velocity
SDOF	Single Degree Of Freedom
SGB	Stochastic Gradient Boosting
SSE	Sum of Squared Errors
V_{s30}	the time-averaged shear-wave velocity to a depth of 30 meters

1. INTRODUCTION

Catastrophic events may cause casualties and damage to properties such as buildings, roadways, bridges, and the contents of the assets. These property damages lead to monetary loss for governments, the insurance industry, and people affected by events. Risk assessment and reduction, catastrophic management, and loss models should be developed to mitigate unwanted results from these events. While creating these models, the physical damage of assets under the effects of strong ground motions must be adequately represented.

Fragility functions are known as estimators of potential building damage by representing exceeding a given damage state at different intensity levels. These representative functions of building damage are derived for various typologies with four different methods: professional judgment; empirical method, which desires the collection of building damage data from past earthquakes; analytical method, which requires non-linear time history analysis under earthquake records; and hybrid method, the combination of analytical and empirical methods (Ahmad et al., 2011; Shinozuka et al., 2000; SYNER-G, 2011).

The professional judgment method relies on the information of experts. The damage probability at several intensity levels of ground motion for various structures is expected to be estimated by experts (SYNER-G, 2011). This data may not be affected by actual structural damage data or statistics; however, the estimations' reliability increases when the consulted experts' sample size increases. This method has deficiencies due to reflecting only the opinion of the experts.

The empirical method needs observed damage data from past earthquakes. Even the most reliable and realistic approach while deriving fragility functions is this method; this method has a few drawbacks. The first reason is that undamaged buildings are not recorded after an earthquake. Another reason is expert judgments on classifying

damaged buildings. Some experts assess that a building has slight damage in that condition, and some evaluate it as moderate damage. So, there could be inconsistency when labeling the damage state of the buildings. Moreover, on-the-ground surveys can be insufficient to determine a building's past or last condition. Finally, even though the intensity measures of ground motion can be easily estimated at the location of the buildings, these measures are not directly measured from an earthquake itself. Earthquakes involve epistemic and aleatoric uncertainties, so the exact intensity parameters for a specific location cannot be determined. For example, epistemic uncertainties in the site's soil conditions or aleatoric uncertainties in the source-to-path characteristics hinder the precise knowledge of intensity parameters. This can lead to inaccuracies in damage classification at the target intensity level of the ground motion, which is estimated by utilizing ground motion models. However, observational data represent the fluctuation of a building stock's structural capacities and the soil-structure interaction. Hence, deriving fragility functions with observational data is more rational despite these drawbacks.

The analytical fragility functions are a statistical approach that simulates analysis under real-time earthquake records. When analyzing the non-linear time history (NLTH), the structural behavior beneath the data is ascertained. This method also has advantages and disadvantages, just like other methods. The most disadvantageous thing is that this method is computationally burdensome and time-consuming because of many NLTH analyses. The second drawback is setting up representative models to mimic the existing building stock. Unfortunately, representing the variability of the elements, both non-structural and structural, is almost impossible; therefore, it significantly affects the reliability of analysis results. On the other hand, record-to-record variability is considered when utilizing this method, along with using past earthquake records.

The hybrid fragility functions combine the different methods mentioned above. This method compensates for the lack of observational data and gathers more representative data regarding the structure and ground motion variability and uncertainties.

In the scope of this thesis, the analytical method is considered to derive fragility functions due to the disadvantages of empirical and expert judgment methods to generate a more reliable dataset. By using estimation techniques like maximum likelihood estimation after performing multiple stripe analysis (MSA) (Baker, 2015), cloud analysis (CA) (Jalayer et al., 2014), and incremental dynamic analysis (IDA) (Vamvatsikos and Cornell, 2002) which rely on conducting abundant non-linear time history analysis, analytical fragility functions have traditionally been derived using a two-parameter log-normal distribution function. These methods generally consist of computationally burdensome and time-consuming calculations. In recent times, the increasing accessibility of artificial intelligence has contributed to its widespread adoption, while the application of data science in diverse fields has become more popular. Deep learning algorithms (DL) and machine learning (ML) provide benefits such as decision-making capabilities, the ability to handle complex problems, and improved computational efficiency. So, well-trained artificial intelligence algorithms reduce time-consuming analysis and offer faster solutions by finding patterns in data.

1.1. Objectives and Outline of the Thesis

A new approach to analytically generating fragility functions is presented in this thesis, with emphasis on the applicability and evaluation metrics of algorithms. Furthermore, these fragility curves are compared with conventionally derived fragility functions to demonstrate the practicality of machine learning algorithms in deriving fragility functions.

The main aim of this thesis is to derive fragility functions for generic mid-rise building inventory spread over the Marmara Region by predicting the probability mass function of a discrete engineering demand parameter via five various supervised ML algorithms, namely Random Forest (RF), Stochastic Gradient Boosting Classifier (SGB), Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Decision Tree (DT). In this thesis, the maximum inter-story drift ratio is examined as an engineering demand parameter. While deriving fragility functions, six intensity measures of strong ground motion,

namely cumulative absolute velocity (CAV), peak ground displacement (PGD), spectral displacement ($S_d(T_1)$), peak ground velocity (PGV), spectral acceleration ($S_a(T_1)$), and peak ground acceleration (PGA), are utilized to predict the exceedance probability of discrete maximum inter-story drift ratios. This thesis also aims to derive fragility functions from user-defined inputs. Conventional fragility functions are generated for specific building typologies. In some cases, the necessary information for a fragility function may not be available, or structural information may not be considered when a specific fragility function is desired. This thesis will introduce a user-friendly technique that can be utilized to develop fragility functions using user-defined inputs, bypassing the need for extensive structural analysis and addressing the limitations of the previously mentioned methods.

To achieve these objectives, a generic new dataset is generated by simulating 1,080 buildings and performing non-linear time history analysis under 11 earthquake records with three different scale factors and three different soil conditions for these representative buildings in the Marmara Region. As a result, 106,920 NLTH analyses are performed via the OpenSeesPy framework. In this thesis, for the sake of simplicity, 6- and 8-story reinforced concrete buildings constructed between 1980 and 2000 are chosen to denote mid-rise structures in the Marmara region. The preparation of the dataset and the model of structures are discussed in Chapter 2, while the selection of the earthquake records is discussed in Chapter 2. The application of ML algorithms is described in Chapter 3. Chapter 4 presents fragility functions derived from ML algorithms and comparison with the conventional multiple stripe analysis method. In conclusion, Chapter 5 wraps up the thesis by summarizing the discoveries and presenting ideas for future research.

1.2. Literature Review

The increasing use of artificial intelligence in earthquake risk assessment, such as estimating log-normal parameters for fragility curves or predicting post-earthquake damage, demonstrates the impact of artificial intelligence on earthquake engineering.

In this context, Atici et al. (2023) used synthetic datasets and machine learning algorithms to create a building damage prediction model. The synthetic dataset was generated from representative buildings, 1- to 5-story buildings constructed between 1980 and 2000, in the Marmara region to obtain a reliable and balanced dataset. 3,240 representative buildings were analyzed under 11 different earthquake records. The data is accumulated with analyses by applying two scale factors to the selected earthquake records. The generated dataset was evaluated under the K-Nearest Neighbor algorithms, Support Vector Machine, Random Forest, and Decision Tree. Random Forest algorithm delivered highly positive results regarding the dataset, demonstrating a prediction F_1 score of over 90% throughout the tests.

Stojadinović et al. (2022) presented a quick evaluation of seismic damage using representative data samples and machine learning. This method has been conducted for a post-earthquake damage prediction method. This method aimed to forecast the damage states from small representative observed damaged buildings. 1979 buildings damaged by the 2010 Kraljevo earthquake were utilized to gather the dataset in this study. The researchers utilized the Random Forest algorithm to predict the damage states. The first finding is that information regarding the severity of ground motion is implicitly provided by observable structure damage. Thus, any information about earthquakes and local soil types is unnecessary for the data. The second finding is that the loss estimation of 10% of the portfolio as a representative portfolio with the consideration of less than 20% relative error is sufficient to estimate the whole portfolio with acceptable accuracy.

Ulku et al. (2022) utilized machine learning algorithms to estimate the damage states of the buildings based on imbalanced data collected after the 2020 Elazığ Earthquake. Following the earthquake, local governments and the government conducted extensive street assessments of nearly ten thousand damaged buildings. These structures are divided into four categories: moderate damage, substantial damage, total damage, and emergency demolition. This work primarily aims to categorize damage states using six machine learning algorithms: random forest, Naïve Bayes, decision tree,

support vector machine, k-nearest neighbor, and logistic regression. While gathering the data, V_{s30} values based on USGS Global Vs30 Map Server (Wald and Allen, 2007) were obtained from ELER software (Hancilar et al., 2010). Peak ground acceleration (PGA) was calculated with the GMMs of Akkar et al. (2014), Chiou and Young (2008), Cagnan and Akkar (2010), and Zhao et al. (2008) with the weights of 0.3, 0.3, 0.3, and 0.1, respectively. Moreover, this study shows that predicting relatively fewer damage classes than the others is more challenging when the data is imbalanced. Therefore, many combinations of over- and under-sampling methods are utilized, and Tomek's Link under-sampling method has been given the best confusion matrix. According to the study, the best-performing algorithm is the random forest with Tomek's Link under-sampling improvement in the dataset, with a mean accuracy of 85% and an F_1 score of more than 68%.

Wu and Sarno (2022) used machine learning algorithms to develop state-dependent fragility curves of existing steel moment frames with masonry infills. This study's main goal was to use representative intensity measures of strong ground motion to determine if a building will exceed the target limit state. The use of feedforward neural networks (FNN) to determine the state-dependent fragility curves of current steel moment frames while accounting for masonry infills is shown in this paper. The limited-memory Broyden-Fletcher-Goldfarb-Shanno-quasi-Newton (LBFGS) is utilized for training the dataset. The authors used FNN rather than regression models because FNN makes predictions according to the posterior probability of output. This study selects the inter-story drift ratio as the engineering demand parameter to define limit states. Furthermore, 28 near-fault and 22 far-field ground motion records are utilized with three different scale factors to perform non-linear time history analysis. The dataset obtained from utilizing the highest and the lowest scale factors is selected for the training dataset, and the other remains for the test dataset. As a result, the authors claim that fragility curves derived from the proposed method were sufficient estimations when compared to the conventional incremental dynamic analysis method.

Salmi et al. (2022) developed fragility curves for mid-rise reinforced concrete

frames using six machine learning algorithms, including KNN, DT, NB, logistic, linear, and ridge regression. The output was determined from nonlinear time history analysis done under the records of strong ground motion, which are chosen based on generalized conditional intensity measures methodology (GCIM) and Baker's proposed method (2011). Machine learning models are constructed using input data from the outcomes of MSA analysis, which are MIDRs and ground motion features. Furthermore, the fragility curves derived by utilized machine learning algorithms are compared with those derived by conventional MSA analysis. As a result, Baker's proposed dataset resulted in an imbalanced dataset. However, the dataset gathered by utilizing the GCIM method improves to overcome the issue of an imbalanced dataset. Under the GCIM method, the decision tree algorithm achieves the highest accuracy and the best evaluation metrics. Nevertheless, the authors emphasize that these results cannot be eligible for other structural systems. Another important outcome is that the proposed method reduces time and computational demand compared with the MSA method. Despite a 10 percent difference between the proposed and MSA methods, this method can be utilized when quick risk assessment is needed.

Furthermore, Mangalathu et al. (2020) and Roeslin et al. (2020) evaluated machine learning algorithms to assess the building damage of the collected data after the 2014 South Napa earthquake and the 2017 Puebla-Morelos, Mexico, Earthquake, respectively. These papers focus on the most probable prediction when they use more than one ML algorithm. Both researchers highlight that the random forest algorithm can be more efficient with higher than 65% accuracy than the other selected algorithms. Moreover, Roeslin et al. (2020) claimed that the most compelling features of the model are building location, seismic demand, and building height.

2. PREPARATION OF DATASET AND MODEL OF STRUCTURES

2.1. Introduction

For the derivation of seismic fragility functions, first, the earthquake hazard needs to be characterized and quantified. Then, the vulnerability characteristics of the buildings, which are exposed to the earthquake hazard, should be identified and modeled for structural analyses under earthquake actions. Therefore, it is necessary to clarify the dataset's features before training machine learning algorithms. Considering earthquakes, ground motion intensity measures (GMIMs), such as spectral acceleration (S_a), peak ground acceleration (PGA), spectral displacement (S_d), peak ground velocity (PGV), cumulative absolute velocity (CAV), and peak ground displacement (PGD), are used for each machine learning algorithm dataset, individually. In addition to these GMIMs, soil condition is considered for the dataset with three soil types. Furthermore, as building characteristics, the number of stories, number of spans, span length, story height, column width, column depth, concrete strength, and commercial use of the first floor are taken into account for features of the dataset. One of the engineering demand parameters (EDPs), or structural response features, is utilized concerning the output, which is the maximum inter-story drift ratio.

Two methods exist for gathering datasets to train machine learning algorithms for deriving fragility and vulnerability models by utilizing machine learning. The first method is the field observations of damage following earthquakes, and the second method is the use of analytical models to forecast outcomes. Both methods have advantages and disadvantages (Baker et al., 2021).

The former, known as the empirical approach, is supported directly by observable data but necessitates meticulous data gathering. Buildings at each damage level need to be evaluated similarly in order to obtain an unbiased assessment of the damage

state that a building might experience, given a particular level of ground motion. If only buildings that are more than moderately damaged are examined, or some damage states are relatively higher than others, the observed dataset is considered imbalanced. There are various strategies to address an imbalanced dataset, including over or under-sampling (Ulku et al., 2022; Kaur et al., 2020). Ulku et al. (2022) have demonstrated that imbalanced datasets can significantly increase the complexity of the problem. Although the data reflect the variation in ground motion and building characteristics, the rarity of earthquakes may make it challenging to acquire sufficient data. As a result, collecting data frequently necessitates a sizable time and resource commitment. According to Baker et al. (2021), the analytical method requires that the analytical consequence projections accurately represent practice.

To address this issue of an imbalanced dataset, the generation of a balanced artificial dataset by simulating representative buildings and analyzing them with NLTH analysis under the records of strong ground motion has been proposed to mimic the building stock of the Marmara region and generate a balanced dataset for machine learning algorithms.

The Marmara region has been extremely hazardous because it has not had a major earthquake in more than 200 years. Numerous studies brought attention to the issue of earthquakes larger than 7 in the Marmara Sea following the 1999 events in Duzce province (Murru et al., 2016; Hubert-Ferrari et al., 2000; Parsons et al., 2004; Parsons et al., 2000; Atakan et al., 2002; Erdik et al., 2004; Kalkan et al., 2009; Gülerce and Ocak, 2013). Precautions should be taken ahead of time not only because cities in the Marmara region, especially Istanbul, have hosted many people but also because the area is significant due to the presence of numerous risky buildings. Consequently, this dataset's main features are the region and the characteristics of the buildings where most people live.

2.2. Building Characteristics of the Stock in the Marmara Region

There are many studies about building stock characteristics in the Marmara region. (DEZIM, 2020; Azak et al., 2014; Bal. et al., 2007). Therefore, this thesis benefits from these studies to obtain the distributions of each feature. The joint probability distribution of the year of construction and the number of stories of Istanbul's reinforced concrete buildings is shown in Figure 2.1. In the scope of this study, 6 and 8-story reinforced concrete structures constructed between 1980 and 2000 are selected due to representing more vulnerable buildings to earthquakes in the building stock and covering most of the buildings referred to as mid-rise reinforced concrete structures.

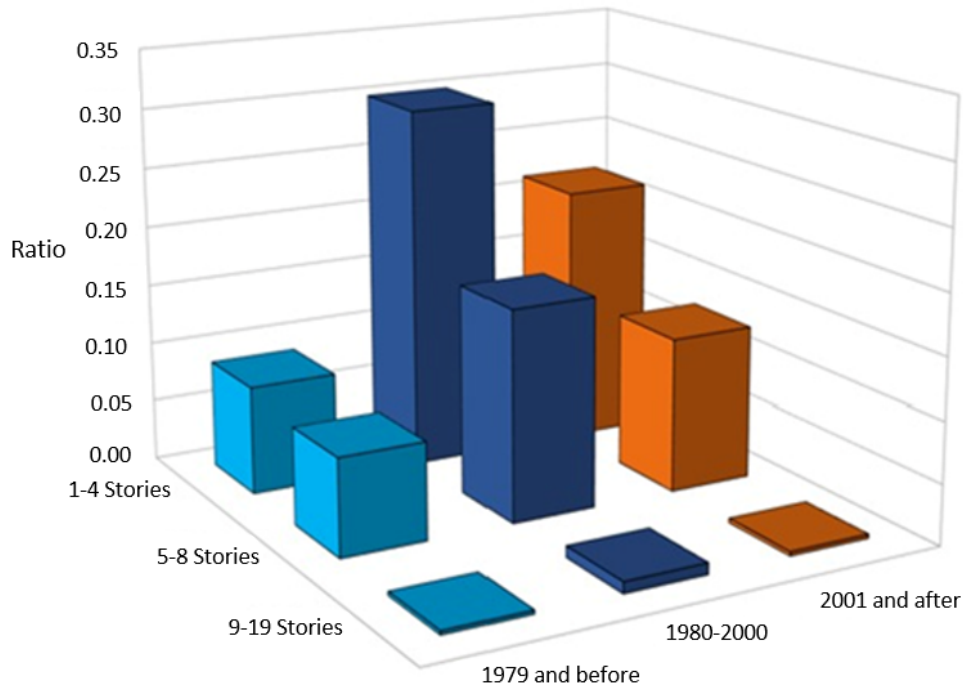


Figure 2.1. Variation of the percentage of the stock related to the construction year and number of stories of reinforced concrete frame buildings in Istanbul (DEZIM, 2020).

In addition to the literature review of the building stock, the distribution of characteristics that could not be found in the previous studies was assumed to meet the minimum regulatory requirements of the Türkiye Building Earthquake Code (ABYYHY, 1975) published in 1975 due to the consideration of the buildings constructed between 1980 and 2000.

Table 2.1. Characteristics of the representative buildings, references, and the variations of the characteristics.

Characteristics of Buildings	References	Variations	Number of Variations
Number of Stories	DEZIM (2020)	6 and 8	2
Story Height	Bal et al. (2008)	2.6, 2.8, and 3 m	3
Commercial Use of Ground Floor	Bal et al. (2008) and Azak et al. (2014)	‘Yes’ or ‘No’ If ‘Yes’, Ground Floor Height = 3.5 m	2
Number of Spans	Azak et al. (2014)	4 and 5	2
Span Length	Bal et al. (2008) and Azak et al. (2014)	2.5, 3, and 4 m	3
Column Dimensions	Azak et al. (2014)	6 story: 30, 35, 40 cm 8 story: 40, 45, 50 cm	3
Beam Width		25 cm	1
Beam Depth	Bal et al. (2008)	60 cm	1
Concrete Comp. Strength	Bal et al. (2008)	5, 12, 20, 28, and 35 MPa	5
Yield Str. of Steel	Bal et al. (2008)	370 MPa	1
Reinforcement Ratio for Column	ABYYHY-1975 (Minimum Requirement)	0.01	1
Reinforcement Ratio for Beam	ABYYHY-1975 (Minimum Requirement)	0.004	1
Total Number of Variations (Number of distinct buildings in this thesis)			1,080

The characteristics of buildings, which are the number of stories, story height, commercial use of the ground floor, the number of spans, span length, column depth, column width, beam depth, beam width, the compression strength of concrete, the yield strength of steel, the reinforcement ratio for columns, the reinforcement ratio for beams, are considered in the scope of the thesis, are shown in Table 2.1 with the variation of the considered values for each one. Consequently, the OpenSeesPy framework (McKenna et al., 2010) was used to simulate 1,080 sample moment-resisting 2D bare frame reinforced concrete.

2.2.1. Physical Properties of Representative RC Buildings

The results of Bal et al. (2007) and Azak et al. (2014) indicate that the standard story height can be defined with log-normal distribution with a mean of 2.84 m and a coefficient of variation (CoV) of 0.08 and 2.71 m and a standard deviation of 0.39 m, respectively. However, the ground floors of many residential and commercial buildings are used for commercial purposes. Therefore, the first story can be less stiff than the standard story. This can lead to the development of soft-story irregularity. This thesis considers commercial use of the ground floor by taking the story height as 3.5 m. As a result of these findings, story height varies between 2.6, 2.8, and 3 m. Nevertheless, while preparing the dataset for machine learning algorithms, the distribution of the story height is considered for standardization.

Figure 2.2 demonstrates the distribution of standard story height by statistically evaluating 938 sample buildings in Bal et al. (2008), and Figure 2.3, the box plot, shows the upper quartile and mean values of the ground and standard story height considered 33,773 examined buildings in Azak et al. (2014).

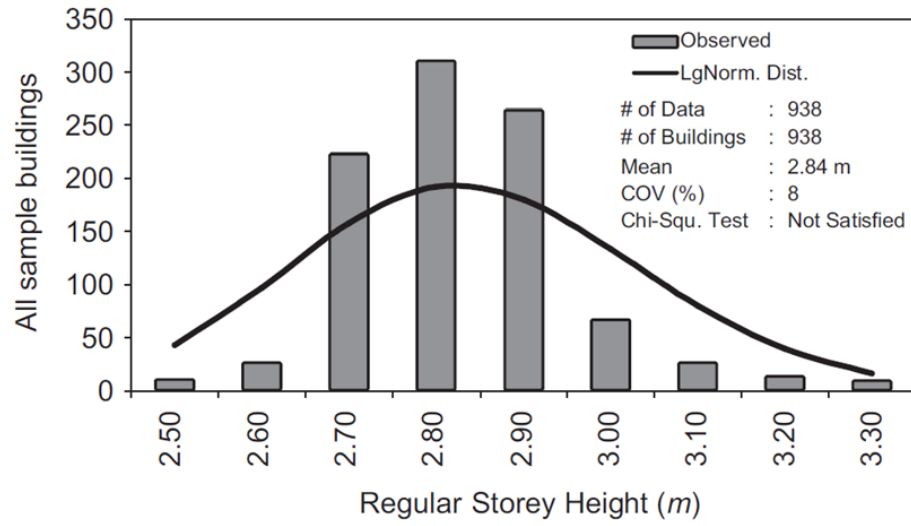


Figure 2.2. The distribution of standard story height for all RC buildings (Bal et al., 2008).

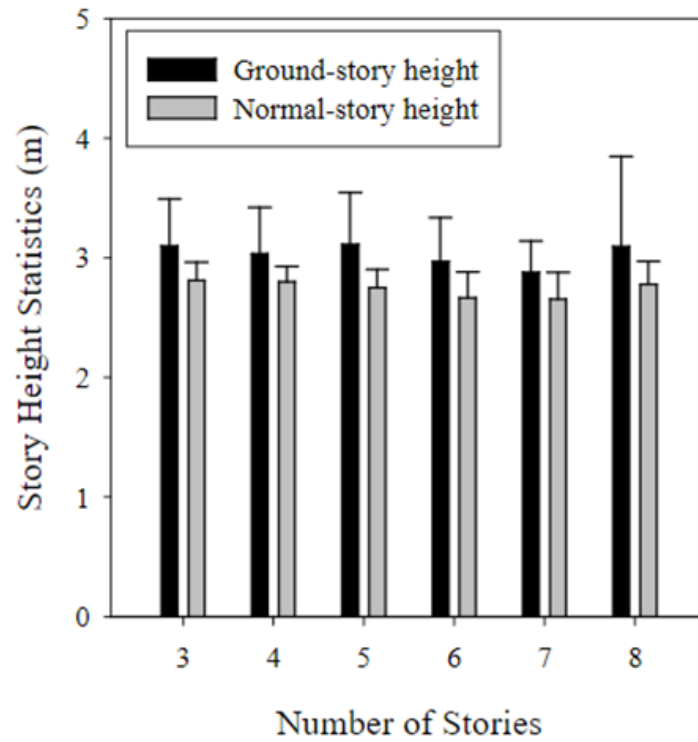


Figure 2.3. The box plot of standard and ground story height for all RC buildings (Azak et al., 2014).

Another essential characteristic of the representative buildings is span or beam length. The distribution of this feature was also mentioned in Azak et al. (2008) and Bal et al. (2014). According to their research, when the orientation is disregarded, Bal et al. (2008) indicate that gamma distribution is satisfied with a mean of 3.37 m and a coefficient of variation of 0.38, similar to the finding of Azak et al. (2014), which the mean of span length is 3.55 m with the standard deviation of 0.68 m. Thus, the span length is considered with three different variations, which are 2.5, 3, and 4m, to provide the given distribution of the span length in Figure 2.4.

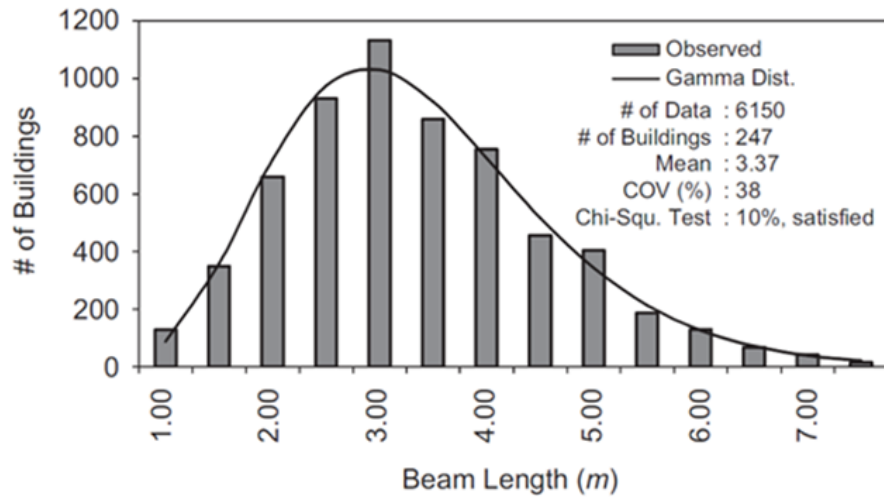


Figure 2.4. The distribution of span length for all RC buildings (Bal et al., 2008).

Azak et al. 2014 also mentioned the distribution parameters of the short plan dimension of 6-story buildings with a mean of 11 m and a standard deviation of 3 m. Moreover, the study also indicates that the mean and standard deviation of the distribution parameters for the long dimension are 15 and 6, respectively. Therefore, when the 6-story buildings are considered for the number of spans, the mean can be considered as with the mean of 13, and the standard deviation of 3.35 m regarding the plan is square. A similar calculation is done for 8-story buildings, which is approximately the mean of 15 m and the standard deviation of 3.2 m. Figure 2.5 illustrates the distribution parameters of the short and long dimensions of various numbers of stories.

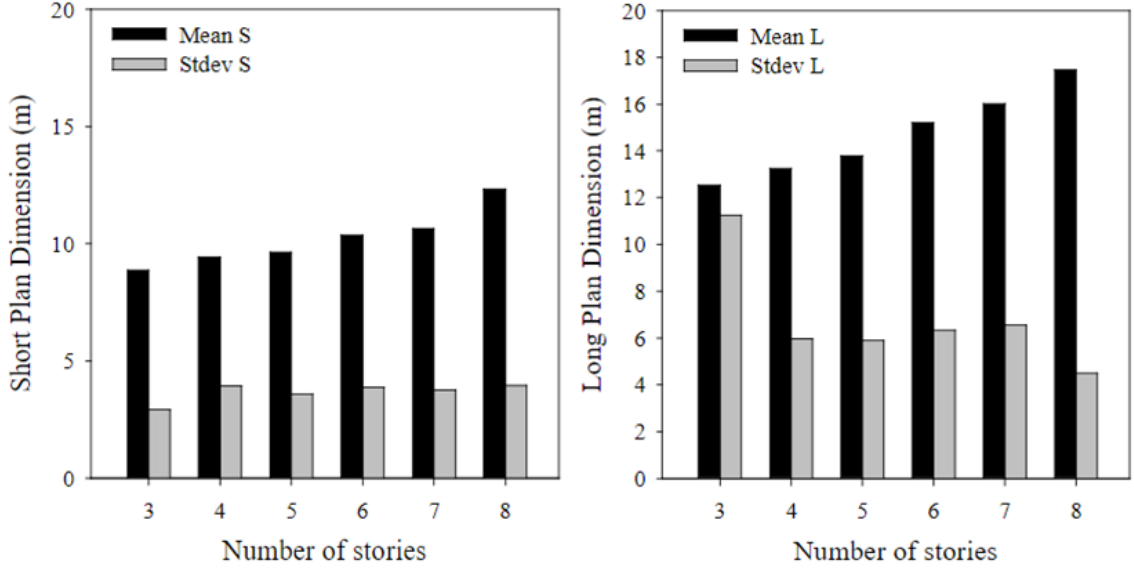


Figure 2.5. The floor plan directions for the short (left panel) and long (right panel) have their mean values and matching standard deviations. (Azak et al., 2014).

As a result of the number of spans of 6- and 8-story buildings with the consideration of beam length as 2.5, 3, and 4, the number of spans is taken into account in the model as either 4 or 5 spans.

Columns are considered to be square for simplicity. Column dimensions are taken into consideration with the condition of the number of stories based on the statistical examination of Istanbul building stock by Azak et al. (2014). Figure 2.6 and Figure 2.7 illustrate the box plot of column width and depth with first and third quantiles with together mean and ± 1 standard deviation, respectively. The box plots indicate that for six-story buildings, the column width and depth mean, minimum and maximum bounds, and dimensions are, respectively, 26, 43, and 34 cm. Thus, the thesis takes into account that the dimension of columns varies as 30, 35, and 40 cm for 6-story buildings in the finite model. On the other hand, the dimensions of columns of 8-story buildings are considered 40, 45, and 50 cm because the weight of buildings is going to increase.

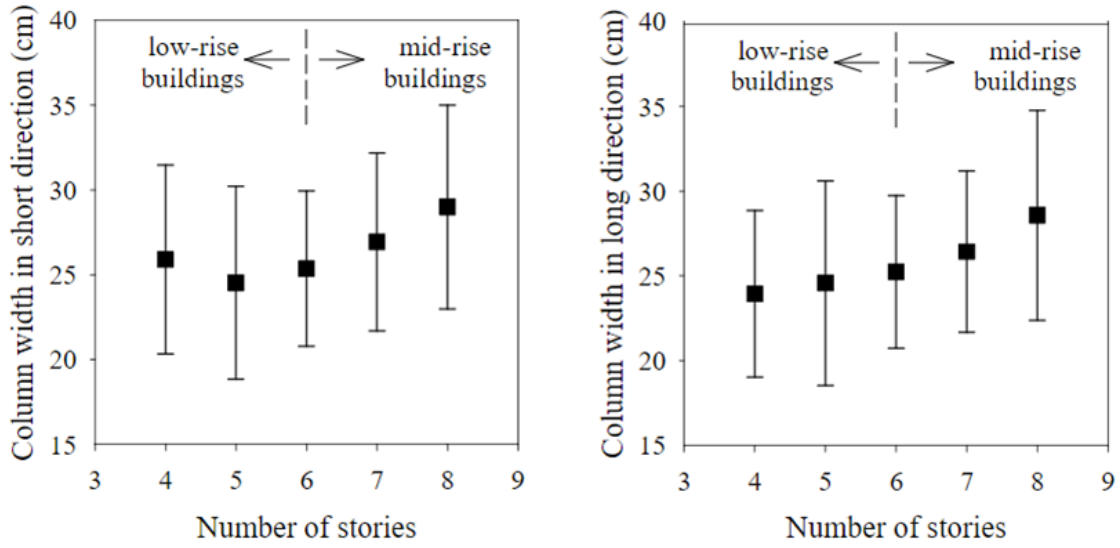


Figure 2.6. Statistics for average column widths of the floor plan according to story number in the short (left panel) and long (right panel) directions (Azak et al., 2014).

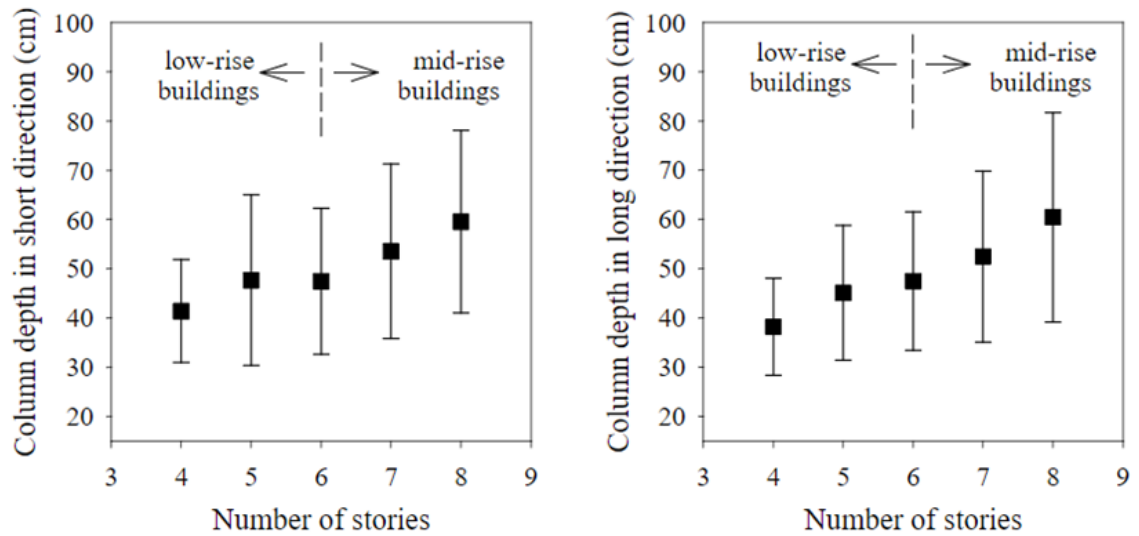


Figure 2.7. Statistics for average column depth according to story number in the floor plan's short (left panel) and long (right panel) directions (Azak et al., 2014).

Another essential characteristic of a building is related to beam dimensions. Bal et al. (2008) obtained statistical knowledge of beam depth from 6,150 examined beams

from 247 different buildings. According to the results of these examinations, the mean depth of buildings constructed before 2000 is 0.6 m. However, there is no strict statistical information about beam width; thus, this study accepts the minimum requirements of Turkish building regulations published in 1975 (ABYYHY, 1975) for beam width, which is 25 cm.

2.2.2. Material Properties of Representative RC Buildings

The study conducted by Bal et al. (2008) is used to calculate the yield strength of steel and the compression strength of concrete. According to the study, the distribution of the concrete compressive strength was generated by considering 6076 core samples from 1178 buildings. This study also claimed that the majority of these buildings were constructed with concrete mixed in the site before being poured and laid. The quality of concrete can be deficient in some cases due to this reason. The study revealed that the average value of concrete compression strength was approximately 17 MPa with a standard deviation of roughly 8 MPa. The suggested distribution of concrete compression strength is the gamma distribution, which is fitted with truncation between 2 and 40 MPa to the samples. Figure 2.8 illustrates the fitted gamma distribution to the samples of concrete compression strength. Concrete compressive values were taken into consideration in five variations in order to depict this distribution appropriately. The values are 5, 12, 20, 28, and 35 MPa.

In 1997, Türkiye's biggest steel mills ceased manufacturing S220-type steel. Nevertheless, before this year, mainly S220 steel was used. Figure 2.9 demonstrates the distribution of S220 steel obtained from Bal et al. (2008). However, the data gathered by Bal et al. to generate this figure originates from the published studies by Akyuz et al. (1992). The mean value of the distribution of steel yield strength was 371 MPa with a coefficient of variation of 24%. In this thesis, the distribution was not taken into account to make it simpler. The steel yield strength is 370 MPa, which is very close to the mean of distribution given in these studies.

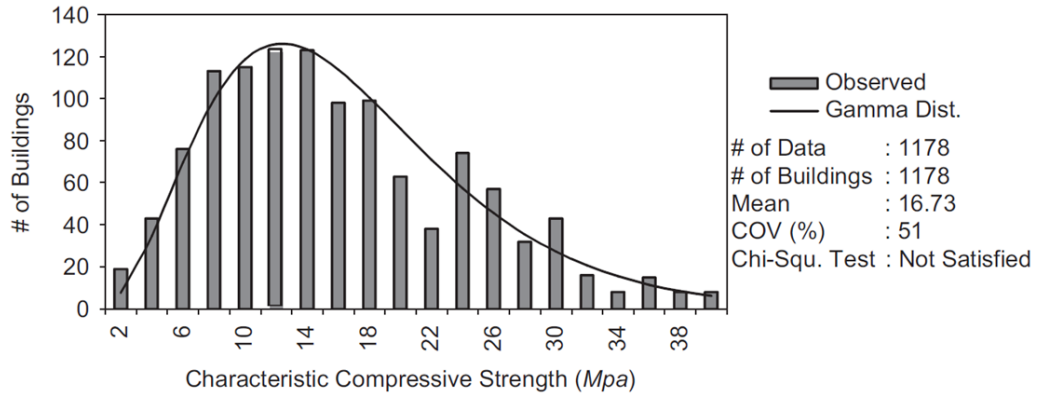


Figure 2.8. Distribution of characteristic 28th-day compressive strength of the concrete sample taken from the Marmara region (Bal et al., 2008).

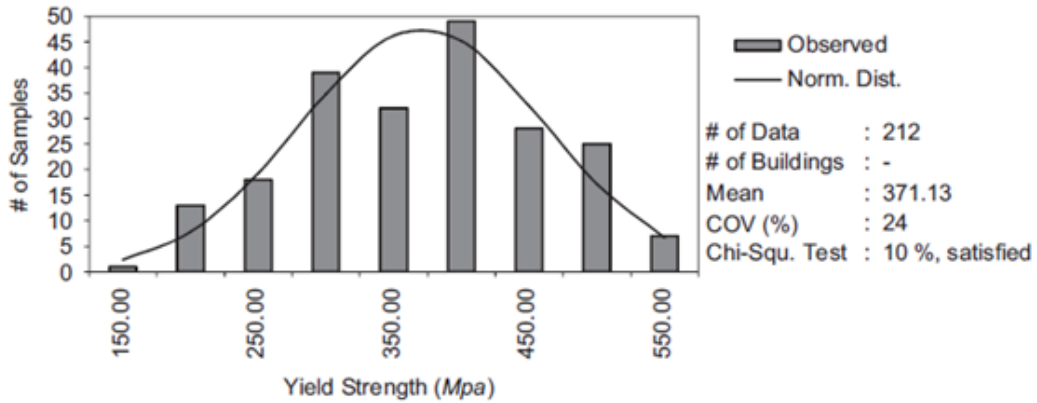


Figure 2.9. Yield strength distribution of S220 reinforcing steel (Bal et al., 2008 and Akyuz et al., 1992).

The reinforcement ratio in columns and beams is established to satisfy the minimal requirements for reinforcement ratio and layout provided in ABYYHY (1975) but is not determined by statistical research. The four corners of the square column and beam sections have been allotted an equal distribution of steel material fibers. A ratio of 0.4% of reinforcement has been employed for beams and 1% for column sections. Section 2.3 of the thesis explains the concept of fiber and the reinforcement configuration in comprehensiveness.

2.3. Finite Element Models of the Representative Buildings

Finite element models of 1,080 representative residential buildings in the Marmara Region are generated as two-dimensional (2D) to simplify the three-dimensional (3D) structures, taking the central axes of them in the OpenSeesPy framework (McKenna et al., 2010). The code is published through the GitHub repository given in Appendix A.

The finite element models are created with the values of building characteristics given in Section 2.2. It was assumed that all the structures' columns and floor plans were square to simplify matters. The materials were assigned “Concrete04” and “Steel01” for concrete and reinforcing steel. The reason for choosing “Concrete04” as a concrete material is to adopt the stress-strain model proposed by Mander et al. (1988). Figure 2.10 illustrates the strain-stress model for unconfined and confined concrete under monotonic loading (Mander et al., 1988). In consideration of the practice of construction of representative buildings constructed between 1975 and 2000, concrete material was thought of as unconfined concrete. Concrete strains were assumed as 0.002 and 0.005 at maximum and crushing strengths, respectively. Moreover, the elasticity modulus is calculated by utilizing Equation 2.1, which was proposed by Mander et al. (1988).

$$E_c = 5000 \times \sqrt{f_c} \text{ MPa} \quad (2.1)$$

where E_c is the elasticity modulus and f_c is the 28th-day compressive strength of concrete.

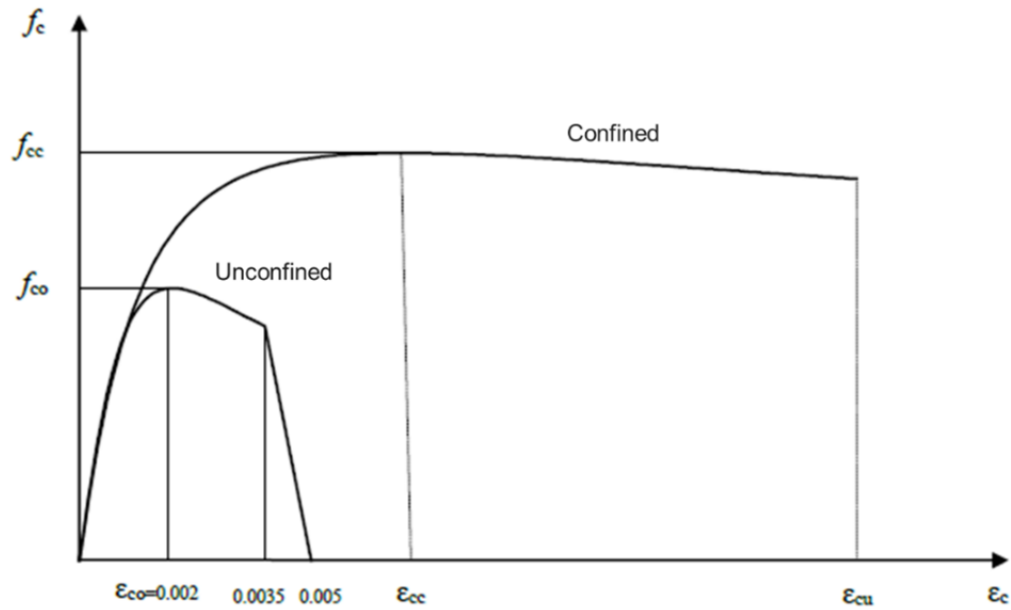


Figure 2.10. The strain-stress models for unconfined and confined concrete under monotonic loading (Mander et al., 1988).

The characteristics of reinforcing steel are given in Table 2.2. As mentioned before, S220 reinforcing steel was used for reinforcement. The yield strength of steel is 370 MPa, while the ultimate strength is 444 MPa. Moreover, the strain-hardening ratio, the ratio between post-yield tangent and initial elastic tangent, was accepted as 3.4×10^{-3} . Figure 2.11 demonstrates the hysteretic behavior of the model used in the OpenSeesPy framework suggested by Spacone and Filippou (1996). The elasticity modulus of the reinforcing steel is used as 2×10^5 MPa in the scope of this thesis.

Table 2.2. Values of the yield strength, strain hardening ratio, and modulus of elasticity for reinforcing steel.

F_y (MPa)	b	E_0 (MPa)
370	3.4×10^{-3}	2×10^5

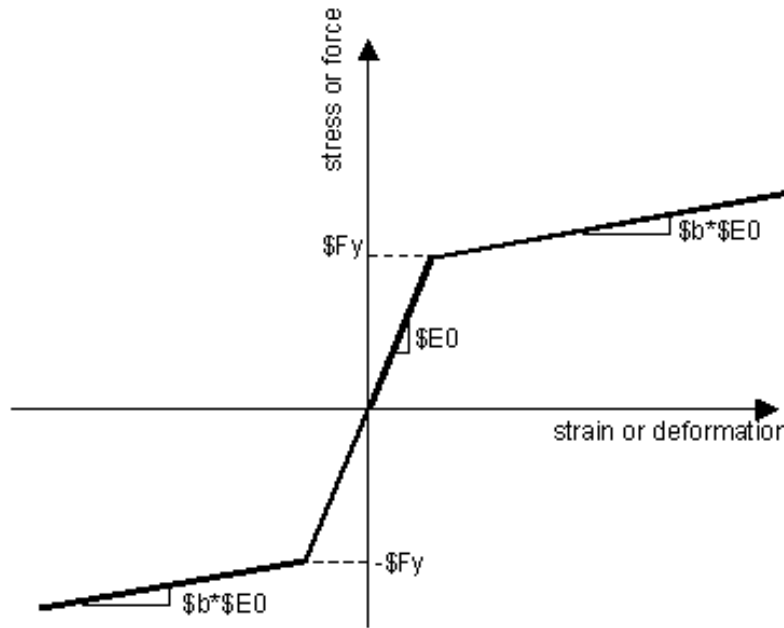


Figure 2.11. Stress-strain model without isotropic hardening (The OpenSeesPy Framework).

All beams and columns were modeled using the “nonlinearBeamColumn” element, which allows us to consider distributed plasticity spread along the elements. While using the “nonlinearBeamColumn” element, each of the beams and columns has five integration points, which means each element is divided into four sub-elements, and for every junction of the sub-element where bending moments are greatest in the absence of inner element loads, internal forces and moments are computed. This method is called the Gauss-Lobatto integration method.

The reinforced concrete section is defined using the “RCSection2d” command for columns and beams. This command allows users to define reinforcement replacement for core and confined regions of concrete. The reinforcement area for the column and beam sections was calculated in Equation 2.2. Half of the total reinforcement is placed at the uppermost portions of the beam and column, while the other half is placed at

the bottom of the sections.

$$Section_{depth} \times Section_{width} \times reinforcement\ ratio = Reinforcement\ Area \quad (2.2)$$

In addition to these details for columns and beams, the P-Delta effect is considered in columns. The P-Delta effect, also defined as secondary moments, appears in cases when axial load and displacement occur concurrently. This leads to the formation of a secondary moment, which is essential to consider when there is large axial force and lateral displacement. The P-Delta effect for an element is shown in Figure 2.12.

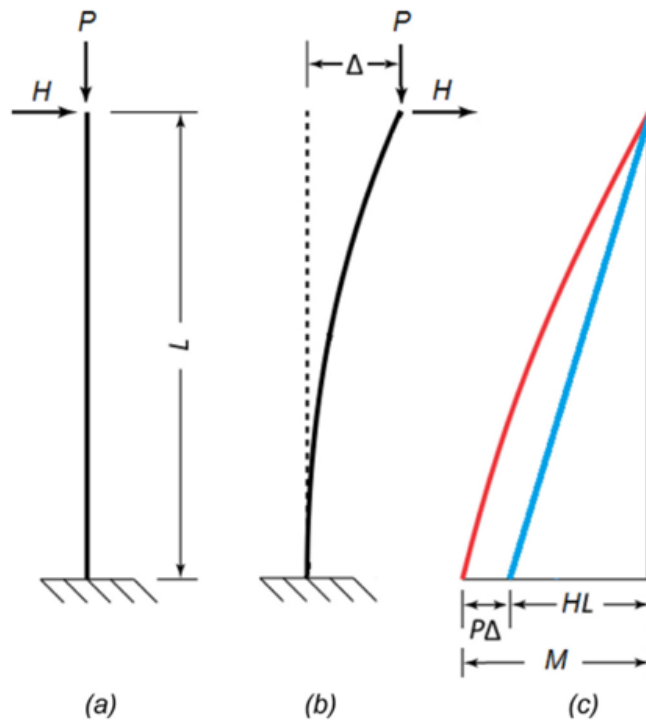


Figure 2.12. P-Delta effect: (a) undeformed configuration, (b) deformed configuration, (c) bending moment distribution (Burgos and Silva, 2023).

The columns are fixed to the foundation for three degrees of freedom. There are 3 degrees of freedom because the model is established in two dimensions. The mass of beams and columns are considered in the model by applying the masses to the elements uniformly distributed. The concrete density is 23.5 kN/m^3 , and the density of the walls

in the inner and external components is considered to be 2.5 kN/m^2 . This wall density is applied along to beams as load. Moreover, the slab's weight is also applied to the beams by considering its thickness, which is 15 cm , and loading tributary area, as shown in Figure 2.13. Superimposed dead load and live load are assigned to the model by considering 2 kN/m^2 . These loads are also assigned to the beams by calculating the tributary area, which is similar to the slab's weight. For seismic analysis, live loads are multiplied by a dynamic load mass participation coefficient of 0.3 for residential-type buildings defined in the 1975 Seismic Code, in addition to dead loads ($G + 0.3Q$). The front view of the two models of representative buildings that were created is illustrated in Figure 2.14.

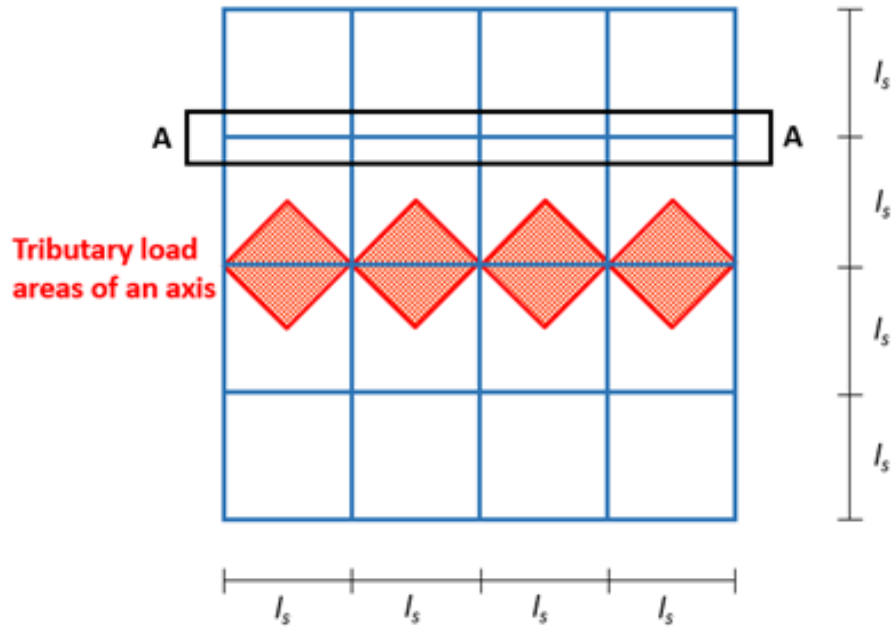


Figure 2.13. The plan of the 4-span building and tributary loads for beams.

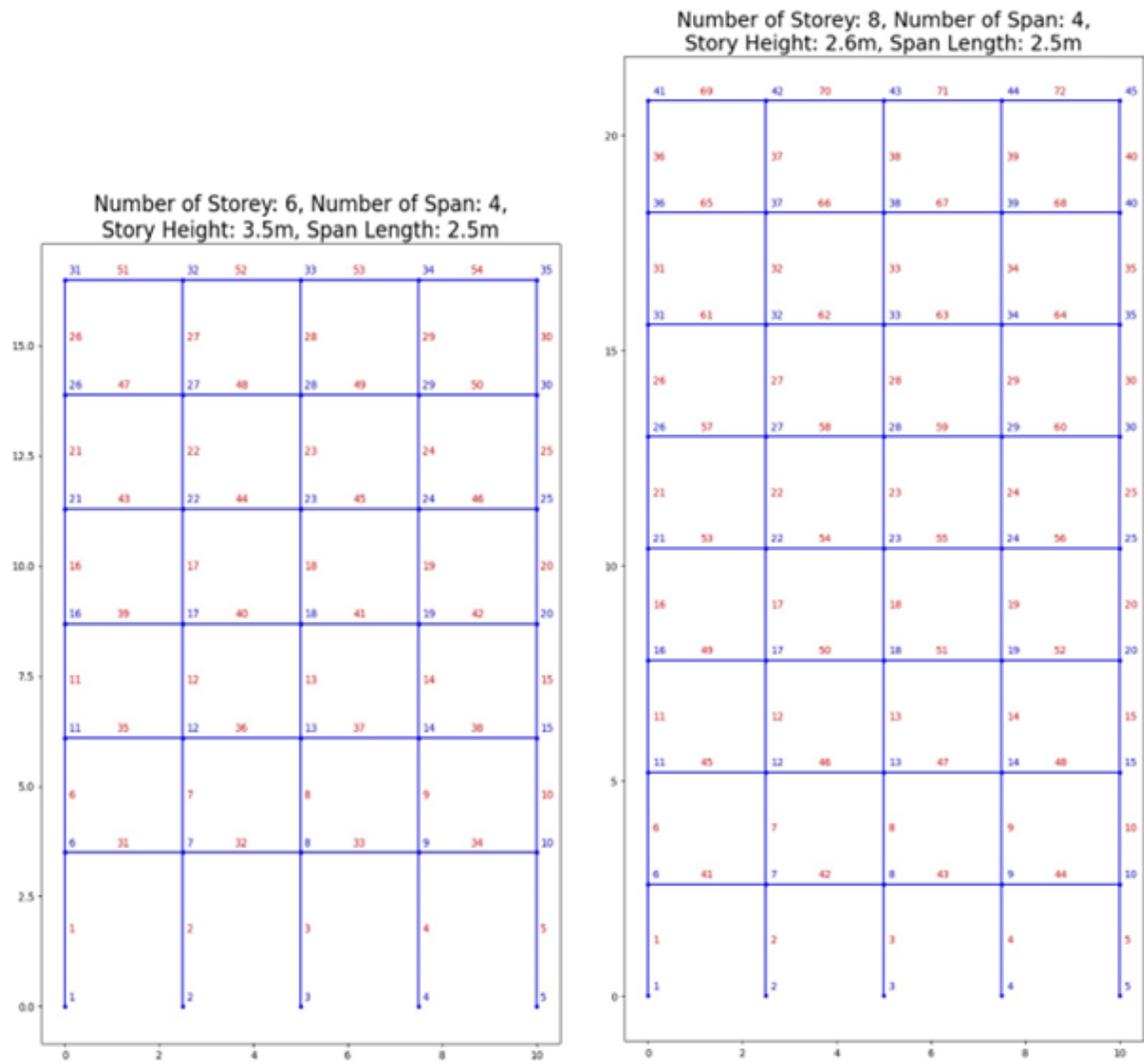


Figure 2.14. Examples of models generated via OpenSeesPy for 6- and 8-story representative buildings.

2.3.1. Verification of the Models

OpenSeesPy models were validated by modeling the same 2D models in the SAP2000 (Computer and Structure Inc., 2000) program, and two models generated from different frameworks were compared with each other in order to verify that loads were assigned, and assumptions were made in the same conditions. To understand that the natural vibration periods and modal participation mass ratios were compared. All assumptions and loads were found to be similar due to the consistency in comparisons. Table 2.3 demonstrates the comparisons of one of the models' modes.

Table 2.3. The comparison of OpenSeesPy and SAP2000 Models.

Number of Mode	Period (second) (OpenSeesPy)	Period (second) (SAP2000)	Participating in Mass Ratio (OpenSeesPy)	Participating Mass Ratio (SAP2000)
1	0.6027	0.6028	0.852 (UX), 0.100 (RY)	0.852 (UX), 0.111 (RY)
2	0.2025	0.2025	0.100 (UX), 0.497 (RY)	0.100 (UX), 0.488 (RY)
3	0.1224	0.1224	0.030 (UX), 0.041 (RY)	0.030 (UX), 0.040 (RY)
4	0.0912	0.0912	0.012 (UX), 0.051 (RY)	0.012 (UX), 0.050 (RY)
5	0.0758	0.0758	0.005 (UX), 0.006 (RY)	0.005 (UX), 0.006 (RY)
6	0.0683	0.0683	0.001 (UX), 0.005 (RY)	0.001 (UX), 0.005 (RY)
7	0.0635	0.0635	0.791 (UZ)	0.791 (UZ)
8	0.0540	0.0540	0.238 (RY)	0.238 (RY)
9	0.0446	0.0446	0.078 (UZ)	0.078 (UZ)
10	0.0361	0.0361	0.018 (RY)	0.018 (RY)

2.4. Assessment of the Structural Response of the Models

NLTH analyses were conducted utilizing the OpenSeesPy framework to determine structural response under actual strong ground motions. Therefore, ground motion selection is vital to assess the seismic behavior of structures. Taking into account the

aforementioned information, ground motion recordings were chosen using Baker and Lee's (2018) conditional mean spectrum approach. This method and the general idea underlying the selection of ground motion records are thoroughly mentioned in Section 2.4.1.

The intensity measure is a parameter used to determine the impact of ground motion on a structure. Time domain parameters, response spectral parameters, and energy-related parameters, which are intensity measures of strong ground motion, are plainly mentioned in Section 2.4.2.

The damage triggered by strong ground motion can be evaluated by Engineering Demand Parameters (EDP). In the scope of this thesis, the maximum inter-story drift ratio was used as an EDP to quantify the dynamic response of a structure under the selected ground motion records. The MIDR and how it is calculated are explained in Section 2.4.3.

The models were examined firstly by static loads and then by non-linear time history analysis under the ground motion excitation. P-Delta effects are considered in the analyses. The periods of free vibration were calculated by utilizing the Eigenvalue. After determining the periods of the modes of models, the modal damping of the 1st and 2nd modes is set to 5% using the Rayleigh damping method since these modes have the highest mass participation. The damping matrix is defined as the result of the combination of the stiffness and mass matrices:

$$c = a_0 \times \mathbf{M} + a_1 \times \mathbf{K} \quad (2.3)$$

where $\mathbf{M}$ is mass and $\mathbf{K}$ is stiffness matrices. The coefficients of a_0 and a_1 are given in Chopra (2013), and the equations are given below.

$$a_0 = \zeta \frac{2 \times \omega_1 \times \omega_2}{\omega_1 + \omega_2} \quad (2.4)$$

$$a_1 = \zeta \frac{2}{\omega_1 + \omega_2} \quad (2.5)$$

The modal damping ratio with respect to the natural frequencies is given in Figure 2.15. Moreover, the damping ratio of the n th mode can be calculated by the equation given:

$$\zeta_n = \frac{a_0}{2 \times \omega_n} + \frac{a_1 \times \omega_n}{2} \quad (2.6)$$

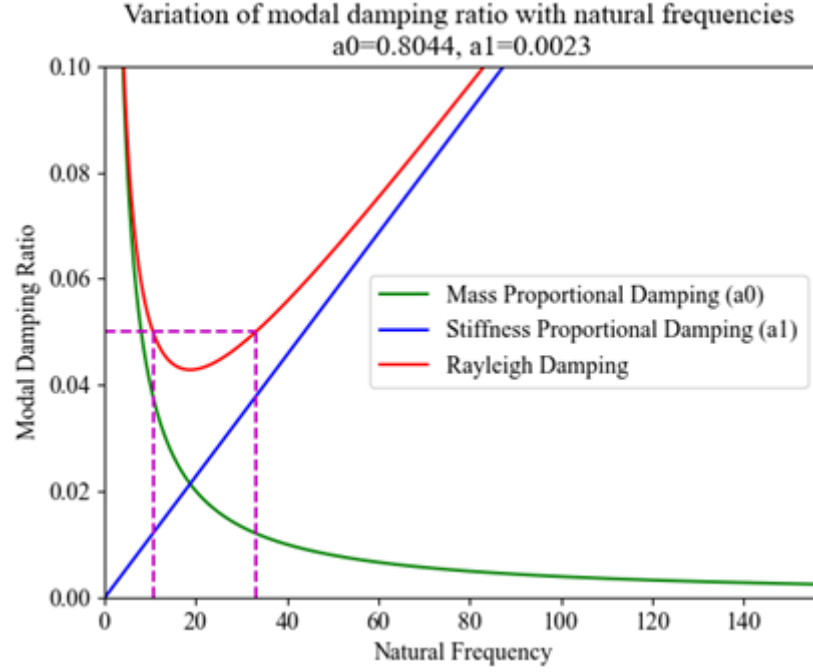


Figure 2.15. Modal damping ratio with respect to natural frequencies for an analyzed sampled model.

The Newmark-Beta method is used as an integration method with the integration coefficients of $\beta=0.25$ and $\gamma=0.5$. The integration method provides to solve the equation of motion by iterating each time step of a ground motion. In some cases, during dynamic analysis, convergence issues are encountered in some models. To address this problem, various solvers have been tried to ensure the completion of the analysis. The process behind the solution is clearly demonstrated in the "dynamic_analysis" function within the "CreateModel.py" file in the GitHub repository provided in Appendix A.

2.4.1. Selection of Ground Motion Records

In this thesis's scope, the CyberShake database (Baker et al., 2021) and a new ground motion database including catastrophic earthquakes that occurred in Türkiye (AFAD) were combined to select 11 different ground motion records for the categorized first vibration period of the buildings. The fundamental vibration period of the generated models is categorized into 10 groups to avoid burdensome and time-consuming calculations (Table 2.4). The reason for categorizing the period of natural vibration is that there will be no substantial difference when using the exact period of natural vibration; however, this shortcut affects the analysis time and keeps fewer ground motion records in the database of the code. 6 out of the 11 ground motion records were obtained from the CyberShake database, while the remaining 5 were taken from the database created within the scope of this thesis, which considered earthquakes in Turkey. This ground motion time-series record selection was conducted by following the improved algorithm of Baker and Lee (2018) by finding appropriate response spectral values under relevant site conditions (V_{s30}) and seismological conditions, such as source-to-site distance and earthquake magnitude. For the sake of simplicity, the seismological conditions were determined based on the consideration of only a single rupture, with the shortest distance from a site to the surface projection of the rupture plane (The Joyner-Boore Distance – R_{jb}) set at a distance of 10 km from the field and earthquake magnitude is 7.56. However, ground motion records are selected according to the source-to-site distance of less than 50 km. Furthermore, when considering the North Anatolian Fault, the faulting mechanism of the fault where this rupture will occur has been determined as a strike-slip fault. Regarding site conditions, the soil class has been considered in three ways for each categorical period. These values represent stiff (ZB), moderate (ZC), and soft (ZD) soil conditions, denoting the time-averaged shear-wave velocity to a depth of 30 meters (V_{s30}) of the site at 1130, 560, and 270 m/s, respectively. ZB and ZC soil conditions show less weathered, medium-solid rock texture; and very tight sand, gravel, and hard clay layers, respectively, while ZD denotes gravel or clay layers (Ince et al., 2018).

Table 2.4. List of period classes and the number of analyses.

Fundamental Vibration Period Class	Period Boundaries (second)	Center of Period Intervals (second) (Considered Period)	Number of Buildings
1	0.35 – 0.45	0.4	55
2	0.45 – 0.54	0.5	202
3	0.54 – 0.64	0.59	270
4	0.64 – 0.73	0.69	221
5	0.73 – 0.83	0.78	154
6	0.83 – 0.92	0.88	102
7	0.92 – 1.02	0.97	49
8	1.02 – 1.11	1.07	17
9	1.11 – 1.21	1.16	7
10	1.21 – 1.31	1.26	3
Total Variety of Building			1,080
Soil Condition			3
Total Selected Ground Motion Records for Each Building and Soil Condition			11
Total Number of Analyses			35,640

Many pieces of research have been done after the 1999 Kocaeli and Düzce earthquakes about the seismic hazard in the Marmara Sea due to the left 150 km unruptured section near Istanbul, which is called the Marmara seismic gap (Hubert-Ferrari et al., 2000; Parsons et al., 2000; Parsons et al., 2004; Murru et al., 2016). Parsons et al. (2000) presented in their work that the probability of earthquake magnitude greater than 7 is 62% with plus or minus 15 in the next 30 years. Moreover, Murru et al. (2016) forecasted the time-dependent probability of an earthquake in the Marmara Sea with the Brownian Passage Time distribution. According to the study, the fifty percentiles of the probability of rupture in the next 30 years if only the Çınarcık segment ruptures is 15.8% with 7.17 magnitude; only the Central Marmara segment ruptures is 17.4% with 7.07 magnitude, and only the West Marmara segment ruptures is 16.6% with 7.17 magnitude. However, in consideration of Kahramanmaraş earthquakes, all three segments can be ruptured together, and its median probability is 1.46% with 7.56

magnitude. According to these studies and considering only one rupture in the thesis, the rupture with magnitude 7.56 is considered.

Baker (2011) presented the conditional mean spectrum with an explanation of the expected response spectrum conditioned on the occurrence of a target spectral acceleration value at the period of interest. Mean log spectral acceleration values conditioned on a period of interest and rupture scenario can be calculated with Equation 2.7.

$$\mu_{\ln Sa(T_i)|\ln Sa(T^*)} = \mu_{\ln Sa}(Rup, T_i) + \rho(T_i, T^*)\varepsilon(T^*)\sigma_{\ln Sa}(Rup, T_i) \quad (2.7)$$

where T^* denotes the conditioning period while $\varepsilon(T^*)$ is a residual quantifying the difference between the conditioning $S_a(T^*)$ and its mean value and can be calculated with Equation 2.8.

$$\varepsilon(T^*) = \frac{\ln Sa(T^*) - \mu_{\ln Sa}(Rup, T^*)}{\sigma_{\ln Sa}(Rup, T^*)} \quad (2.8)$$

The conditional covariance matrix is calculated using Equation 2.9.

$$\Sigma_{cond} = \Sigma - \frac{\Sigma_{cross}\Sigma_{cross}^T}{\sigma_{\ln Sa}(Rup, T^*)} \quad (2.9)$$

Statistically simulated response spectra were generated with the Latin Hypercube Simulation method using the target conditional mean and covariance matrices calculated with the abovementioned equations. After the simulation of response spectra, the sum of squared errors (SSE) is calculated for the simulated response spectrum and each ground motion in the dataset. This ground motion record is selected if the error is less than the maximum tolerance for a candidate ground motion. Otherwise, a further optimization algorithm is applied using a scale factor less than the maximum scale until the error is less than the maximum tolerance. The maximum scale factor was taken as 5, considering the National Institute of Standards and Technology (NIST, 2011).

As a result of the selection of ground motion records, 330 earthquake records were selected considering the 10 classes of the first period, 3 site conditions, and 11 different earthquakes for each site condition and period class. Table 2.5 shows the example of selected records for period class 1 and V_{s30} 1130 m/s. The rest of the selected ground

motions are given in Appendix B. Figure 2.17 illustrates the response spectra of the selected ground motions in both AFAD-TADAS and CyberShake databases for the type of period class and V_{s30} value, which are given in Table 2.5.

Table 2.5. Example of selected ground motions with scale factors and time domain parameters for period class 1 and V_{s30} 1130 m/s.

Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	2703	0.51	0.08	8.4	8.2
Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.3	0.24	40.0	39.8
Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3143	0.6	0.23	74.9	75.4
Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3142	0.5	0.32	43.0	31.2
Ekinozu - Kahramanmaras (02.06.2023)	7.7	North	129	2.16	0.33	40.4	55.4
S. San Andreas; NM+SM+NSB	7.3	1st	s650	3.75	1.02	135.9	160.9
Palos Verdes Connected	8	1st	s313	2.33	0.35	32.2	133.1
S. San Andreas BB+NM+SM NSB+SSB+BG	8	2nd	s123	0.8	0.39	40.1	112.2
Elsinore;W	6.5	1st	PLS	1.72	0.80	42.2	34.6
San Jacinto (SB to C)	7.3	2nd	s564	1.84	0.59	47.6	81.7
S. San Andreas PK+CH+CC+BB+NM+ SM+NSB+SSB	8	2nd	Chino	0.71	0.32	43.8	48.6

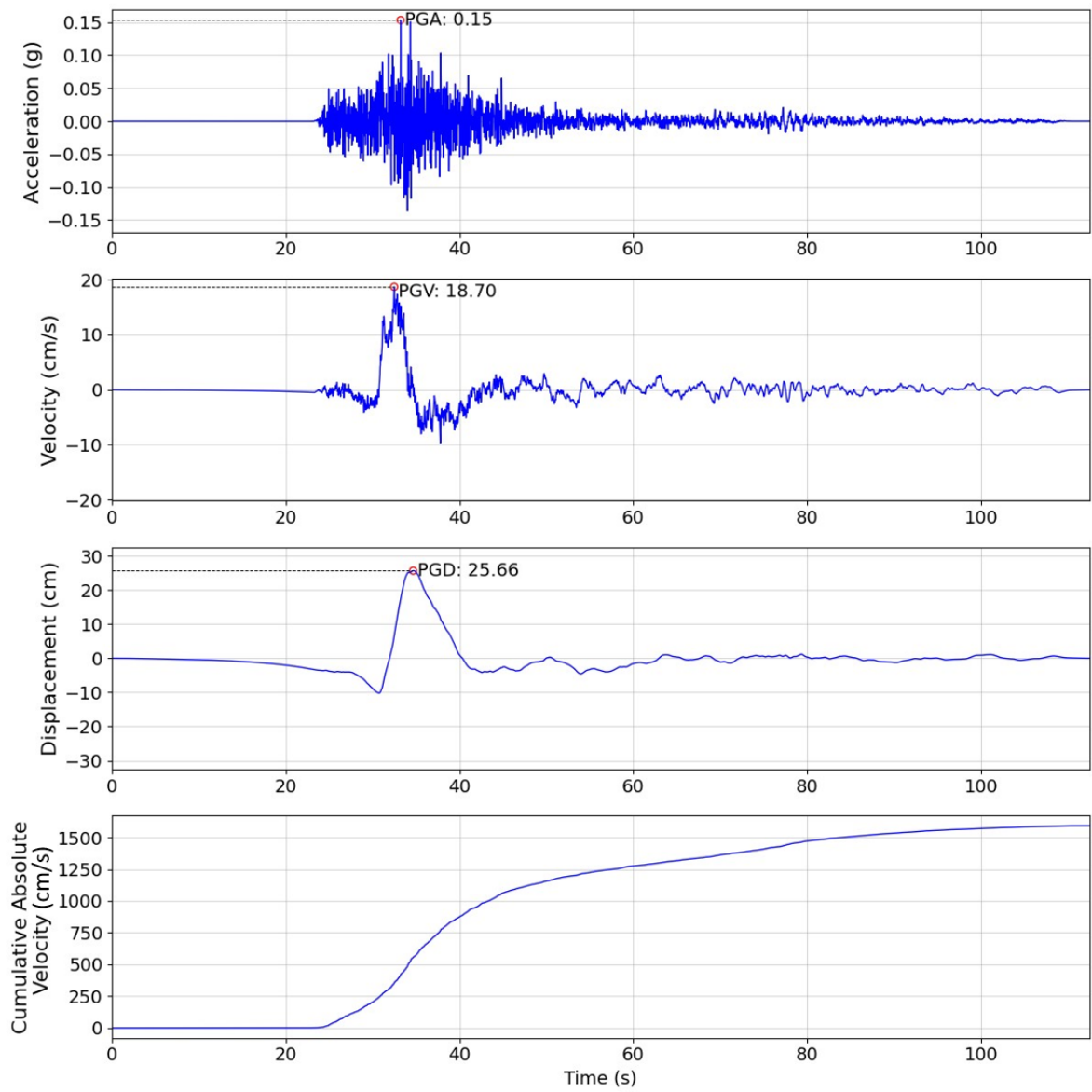


Figure 2.16. Earthquake record obtained from the station 3142 after 02.06.2023
Pazarcik-Kahramanmaras Earthquake.

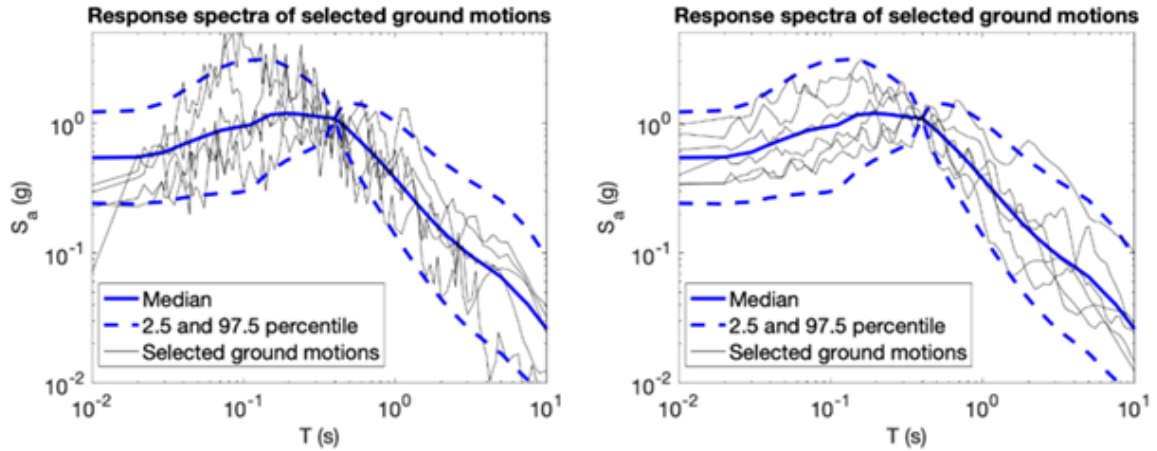


Figure 2.17. Response spectra of selected ground motions for period class 1 and 1130 m/s V_{s30} (left) AFAD database (right) CyberShake database with median conditional mean spectrum (blue solid line) of period class 1 [Figures were plotted by using Baker and Lee's (2018) MATLAB Algorithm].

2.4.2. Intensity Measures of Ground Motion

Intensity indicates the level of shaking induced by strong ground motion at a given location. Intensity varies from location to location due to some parameters such as site condition and source-to-site distance. There are three different types of parameters of intensity. These are time domain (amplitude), spectral (frequency content), and energy-related parameters.

The widely known time domain parameters are peak ground acceleration (PGA), peak ground velocity (PGV), and peak ground displacement (PGD). These parameters can be reached from time histories. The description of peak values is the absolute maximum values of the time series of records. Acceleration is the derivative of velocity, while velocity is the derivative of displacement. So, if there is an acceleration record of an earthquake, velocity and displacement can be calculated by taking the single or double integral of acceleration, respectively. Equations 2.10, 2.11, and 2.12 show how to transform the acceleration record of an earthquake to PGA, PGV, and PGD.

$$PGA = \max(|\ddot{u}_g|) \quad (2.10)$$

$$PGV = \max\left(\left|\int \ddot{u}_g dt\right|\right) = \max(|\dot{u}_g|) \quad (2.11)$$

$$PGD = \max\left(\left|\iint \ddot{u}_g dt dt\right|\right) = \max(|u_g|) \quad (2.12)$$

Response spectrum is an insightful method for defining ground shaking and its impact on structures. Additionally, response spectrum, which is a prominent concept in earthquake engineering, provides an easy way to quantify to peak response of all potential linear single degrees of freedom (SDOF) the maximum response of a single degree of freedom (SDOF) with respect to ground motion by considering the natural frequency and damping ratio of an SDOF systems to a certain component of ground motion. The peak value of a response quantity as a function of the natural vibration period T_n of the system is referred to as the response spectrum for that quantity (Chopra, 2013). The maximum response spectra parameters are spectral acceleration $[S_a(T)]$, spectral velocity $[S_v(T)]$, and spectral displacement $[S_d(T)]$. These three parameters are related to each other. Equations 2.13, 2.14, and 2.15 illustrate the relationship of the parameters. The first step is to calculate the dynamic response by using numerical calculations such as the Newmark-Beta method and central difference method to find the displacement of an SDOF system for each time interval of ground motion (u_t).

$$S_d(T) = \max |u_t| \quad (2.13)$$

$$S_v(T) = \max |\dot{u}_t| \approx \omega_0 S_d \quad (2.14)$$

$$S_a(T) = \max |\ddot{u}_t| \approx \omega_0^2 S_d = \omega_0 S_v \quad (2.15)$$

where ω_0 is the natural frequency of the SDOF system.

There are two different energy-related parameters of ground motion, which are Arias Intensity (AI) and Cumulative Absolute Velocity (CAV). Arias Intensity can be calculated with Equation 2.16, while cumulative absolute velocity can be calculated

with Equation 2.17.

$$I_A = \frac{\pi}{2g} \int_0^{T_d} a_g(t)^2 dt \quad (2.16)$$

$$CAV = \int_0^{T_d} |a_g(t)| dt \quad (2.17)$$

where $a_g(t)$ and T_d are the time history of a ground motion acceleration and the total duration, respectively.

In the scope of this thesis, machine learning algorithms are established for six different intensity measures, and these are PGA, PGV, PGD, $S_a(T)$, $S_d(T)$, and CAV.

2.4.3. Engineering Demand Parameters

Engineering demand parameters (EDPs) are structural response parameters that can be used as a measure of the dynamic response of the structure or to evaluate the damage caused by an earthquake. Commonly used EDPs are the maximum inter-story drift ratio (MIDR), the maximum floor displacement (MFD), the maximum plastic rotations of beams (MPR), the maximum floor acceleration (MFA), and shear force in beam-column joints, axial force, bending moment and shear force in columns, etc. MIDR is used as EDP in this thesis' scope to derive the fragility curves of structures with respect to the intensity measures of strong ground motion.

2.4.4. Evaluation of Dynamic Structural Analysis Results

The first thing that is done is to determine the first vibration period of 1080 representative buildings and classify their first natural vibration period to choose matched ground motion records given in Appendix B. Figure 2.18 illustrates the distribution of the first natural vibration period for representative buildings.

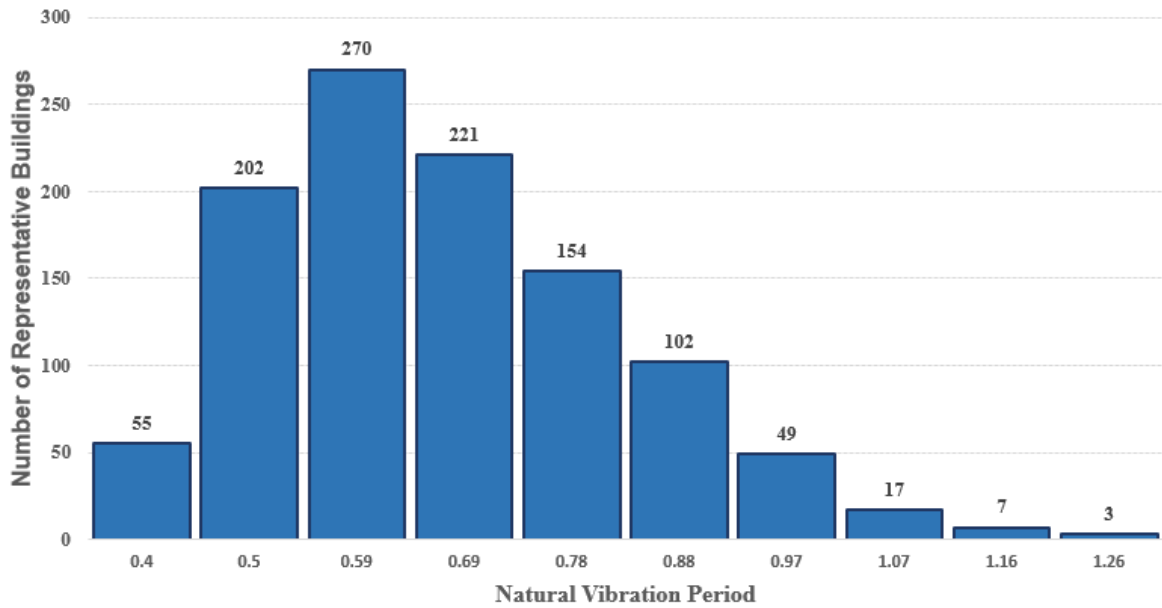


Figure 2.18. The distribution of the first natural vibration period of the representative buildings.

The second process is analyzing 1,080 buildings with non-linear time history analysis under 33 selected ground motion records specific to a period class. As a result, 35,640 buildings were analyzed. Ghobarah (2004) claims that if the inter-story drift ratio for non-ductile moment-resisting frame systems is higher than 1%, it means that the buildings will go to complete damage. Only buildings constructed between 1975 and 2000 are considered; thus, this maximum inter-story drift ratio (1%) is utilized to assign complete damage state to buildings. Moreover, the determination of damage states can be varied due to experts' or researchers' judgment. Therefore, the maximum inter-story drift ratio is divided into 10 discrete values while deriving fragility curves via machine learning. Ground motion is selected and scaled according to a given rupture scenario; thus, generated data is unbalanced data. Figure 2.19 clearly demonstrates that the imbalance resulting from complete damage buildings is much more than the other limits. This leads to a classification problem with the tendency to predict complete damage while developing ML algorithms. To solve this problem, the scale factor obtained during the selection of ground motion is decreased with Equation 2.18, proposed by Wu and Sarno (2022).

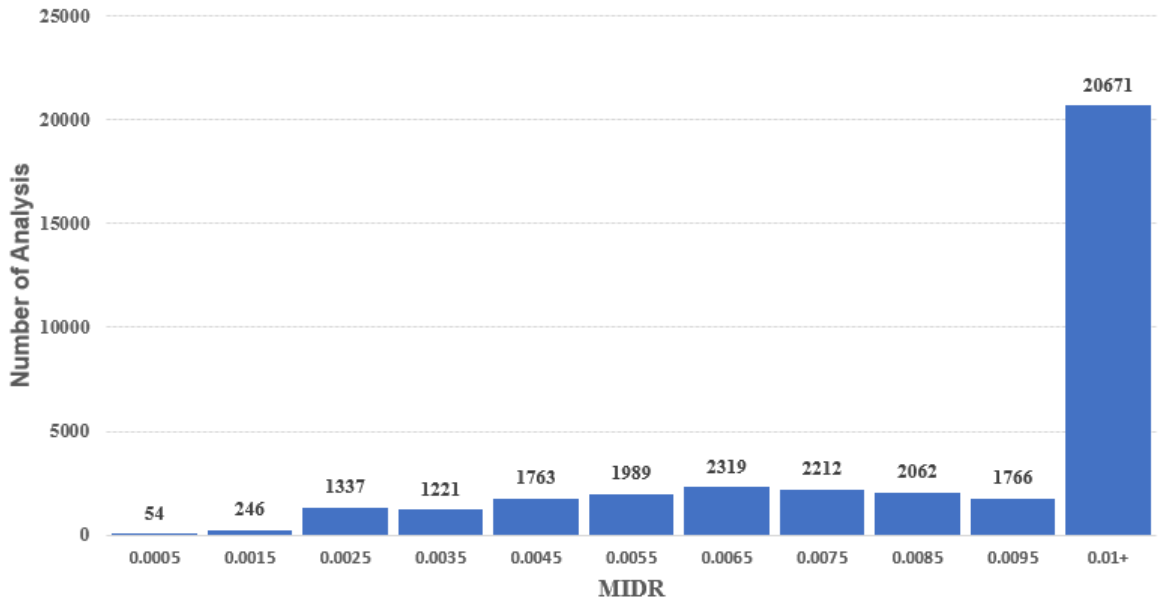


Figure 2.19. The distribution of MIDR of analysis using SF_1 .

$$SF_2 = SF_1 \times \frac{IDR}{\bar{MIDR}} \quad (2.18)$$

where IDR denotes the inter-story drift ratio corresponding to the complete damage limit, which is 1%, while $\bar{MIDR}$ indicates the mean MIDR obtained from non-linear time history (NLTH) analysis while using the first scale factor as the scale factor of ground motion records. $\bar{MIDR}$ is 0.056, which is higher than the complete damage limit.

NLTH analysis is done to calculate the MIDR of the buildings by multiplying the second scale factor with ground motion records. Figure 2.20 shows the distribution of MIDR of analysis using SF_2 .

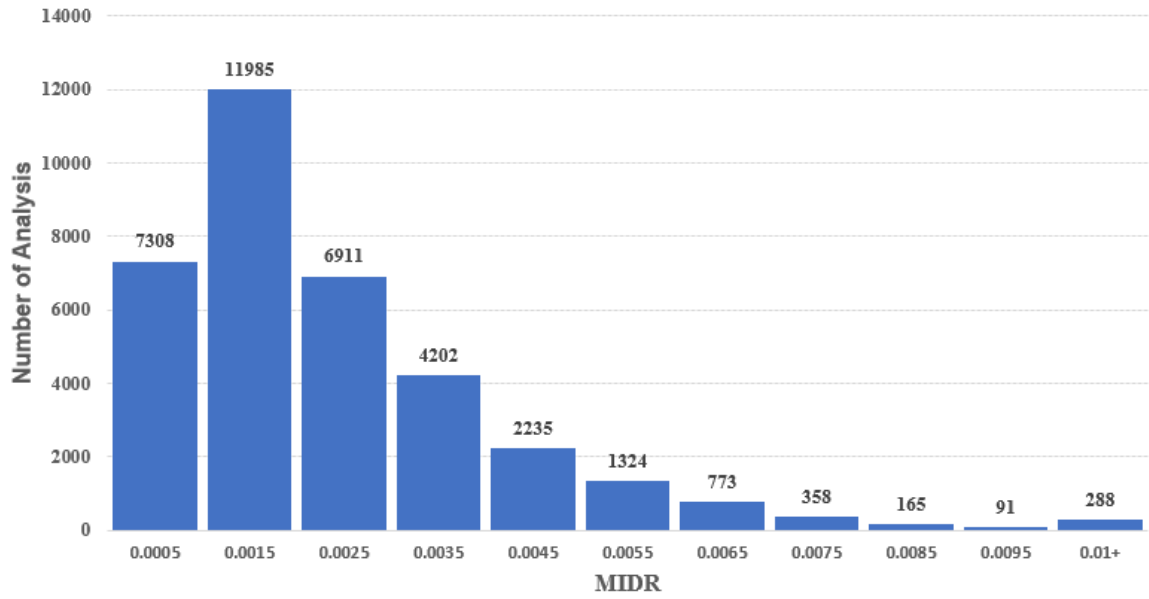


Figure 2.20. The distribution of MIDR of analysis using SF_2 .

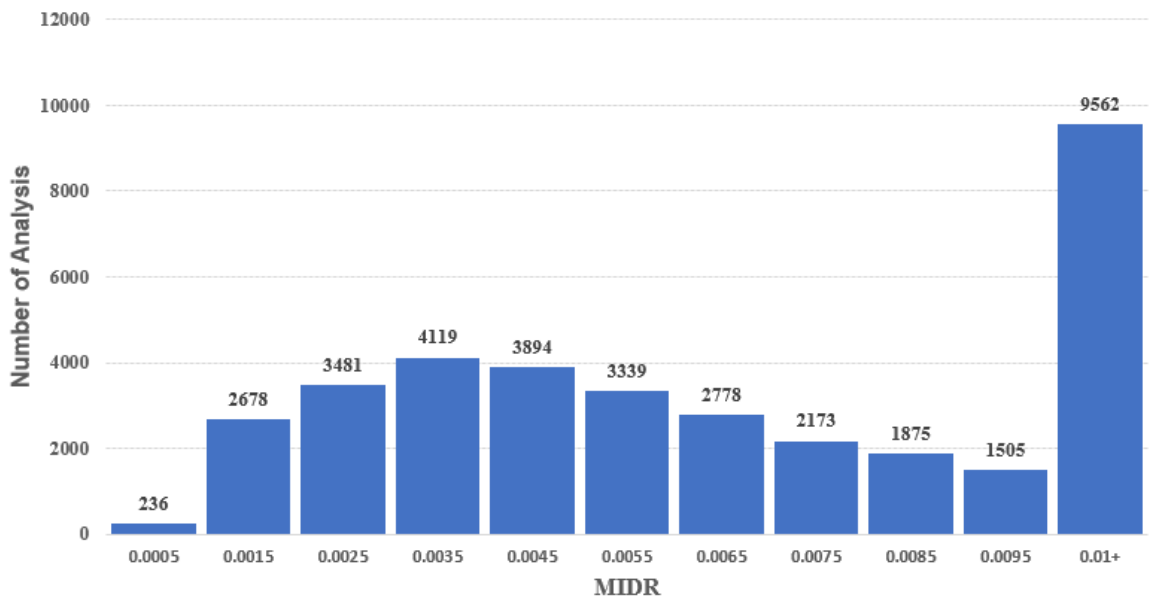


Figure 2.21. The distribution of MIDR of analysis using SF_3 .

Moreover, the analyses were repeated for SF_3 , an arithmetic mean of the SF_1 and SF_2 . Figure 2.21 demonstrates the distribution of MIDR of analysis using SF_3 .

Succeed analyses by utilizing these scale factors are gathered to obtain a dataset for ML algorithms. Figure 2.22 shows the MIDR results obtained from NLTH analyses for different scale factors.

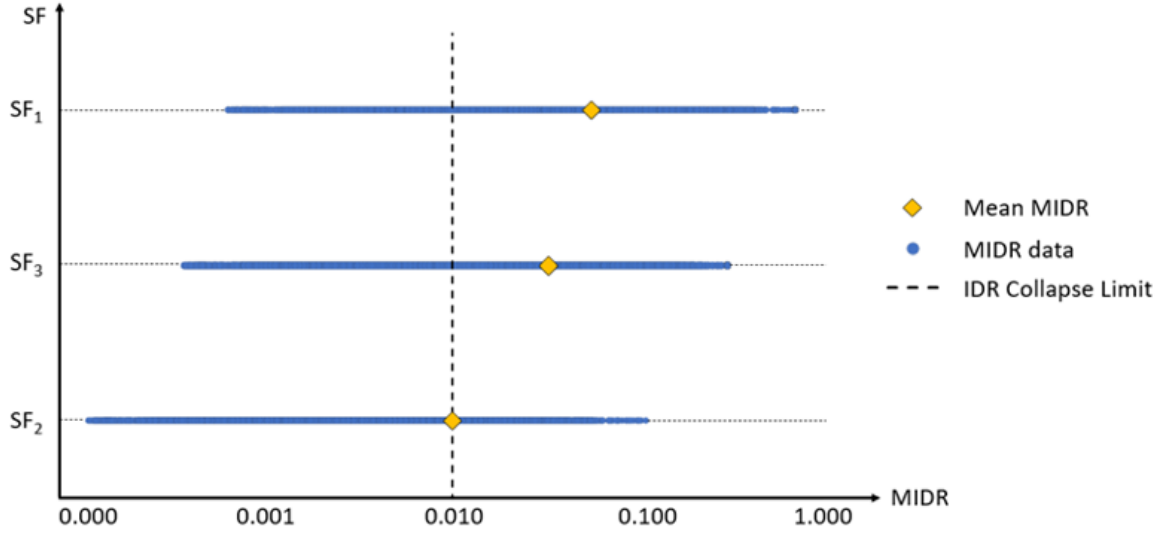


Figure 2.22. NLTH analysis results as MIDR for different scale factors.

At the end of this section, 106,920 analyses were performed to provide a balanced dataset for developing well-trained machine learning models.

3. APPLICATION OF MACHINE LEARNING ALGORITHMS

3.1. Introduction

Machine Learning (ML) algorithms are a way to provide meaning from data or experience. One of the best definitions of the ML algorithms is “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E .” (Mitchell, 1997). In this paper, the classification of the damage level of the buildings is “tasks (T)”. Data preparation is “experience E ”, and the evaluation of whether the algorithms are sufficiently correct or not is “performance (P)”.

There are three types of machine learning applications. One of them is supervised learning, in which the supervisor provides the output or label of data. The second one is unsupervised learning. This kind of application does not have a label or a known output. There are only features. The third one is reinforcement learning. The main idea of reinforcement learning applications is learning from a sequence of actions (Alpaydin, 2014).

In recent times, the increasing accessibility of artificial intelligence has contributed to its widespread adoption, while the application of data science in diverse fields has become more popular. Machine learning (ML) and deep learning algorithms (DL) provide benefits such as decision-making capabilities, the ability to handle complex problems, and improved computational efficiency. The use of these algorithms in deriving fragility curves or damage prediction clearly shows the impact of these on earthquake engineering (Zhang and Burton, 2019; Mangalathu and Burton, 2019; Mangalathu et al., 2020; Ulku et al., 2022; Roeslin et al., 2020; Stojadinović et al., 2021; Kiani et al., 2019; Wu and Sarno, 2022; Atici et al., 2023).

The generated data consists of features [number of stories, number of spans, span length, story height, column width and column depth, concrete compression strength, the purpose of first story usage, soil condition, intensity measure (PGA, PGV, PGD, $S_a(T)$, $S_d(T)$, CAV)] and labels (MIDR level). The raw dataset is shown in Table 4.3 by selecting random data from the database. Therefore, classification algorithms in supervised learning are utilized to predict each building's MIDR level in test data.

Table 3.1. Selected different data from the raw dataset.

Data Label	Input/Output	Data 1	Data 4280	Data 29180	Data 101125
Number of Stories	Input	6	6	8	8
Number of Spans	Input	4	4	4	4
Span Length	Input	2.5	3	3	4
Story Height	Input	2.6	2.6	2.6	2.6
Column Width	Input	0.3	0.4	0.4	0.45
Column Depth	Input	0.3	0.4	0.4	0.45
Concrete Strength	Input	5	12	28	28
Commercial Use	Input	Yes	No	Yes	Yes
Soil Condition	Input	1130	560	560	270
PGA (g)	Input	0.32	0.41	0.55	0.16
PGV (cm/s)	Input	29.8	88.1	86.0	34.5
PGD (cm)	Input	27.5	74.7	77.6	29.3
$S_a(T_1)$ (g)	Input	0.68	1.85	4.16	0.41
$S_d(T_1)$ (cm)	Input	0.96	0.97	4.16	0.59
CAV (m/s)	Input	8.75	15.04	56.12	5.90
MIDR	Output	0.0085	0.0085	0.018	0.003

3.2. Machine Learning Algorithms

The general use of machine learning algorithms in the classification problem is to predict the most probable class. However, in the scope of this paper, the prediction of the probability of each MIDR level for each input is aimed. The dataset is trained by utilizing five various machine learning algorithms, which are Random Forest, K-Nearest Neighbor, Decision Tree, Naïve Bayes, and Gradient Boosting Classifier. These algorithms are explained briefly in the following section.

These machine learning algorithms were implemented using one of the Python open-source libraries, Scikit-learn (Pedregosa et al., 2011). Scikit-learn, a well-known Python package, is an essential tool in the field of machine learning and data science. Scikit-learn offers efficient data analysis and modeling by leveraging a simple yet powerful collection of features, leveraging the capabilities of fundamental libraries such as NumPy, SciPy, and Matplotlib. Additionally, Scikit-learn includes tools that are essential for activities like data preprocessing, feature engineering, and model validation. The library's collaborative and open-source nature, combined with an active community, increases its appeal, making it a vital tool in predictive modeling and statistical analysis within the Python environment.

The general use of machine learning algorithms in the classification problem is to predict the most probable class. Nevertheless, in the scope of this thesis, we aim to predict the probability of damage levels in terms of the range of MIDRs for each input. After predicting the probabilities, fragility curves are derived by utilizing Bayes' theorem and the total probability theorem, which are calculated from Equation 3.1 and Equation 3.2, respectively.

$$P(DL_i = DL | IM = im) = \frac{P(DL_i = DL, IM = im)}{P(IM = im)} \quad (3.1)$$

where DL is a damage level determined by discretizing MIDRs into 10 until the MIDR reaches complete damage, assumed to be 1%.

$$P(DL_i = DL | IM = im_{mid}) = \frac{1}{n} \sum_{i=1}^n P(DL_i = DL | im_{lower} \leq IM \leq im_{upper}) \quad (3.2)$$

where n is the total number of inputs contained in the bin of an intensity measure (IM) while $P(DL_i = DL | IM = im_{mid})$ is the probability of occurring that damage level for each input contained in the bin of an intensity measure.

3.2.1. Decision Tree

One of the non-parametric algorithms is the Decision Tree. The decision tree is a hierarchical model composed of decisions made by recursive splits in the number of steps. A decision tree consists of decision nodes and terminal leaves. (Alpaydin, 2020). Two layers of the fitted best decision tree algorithm are illustrated in Figure 3.1 with impurity measure (Gini index). Split can be measured for a decision of true or false by an impurity measure. If split gives the same classes for each instance at the end of the branches, predicted outputs belong to the same class. This split process will last until all instances are labeled. Equation 3.3 shows the calculations of impurity by utilizing the entropy function (Quinlan, 1986).

$$\mathfrak{J}_m = - \sum_{i=1}^K p_m^i \log_2 p_m^i \quad (3.3)$$

$$\hat{P}(C_i | x, m) \equiv p_m^i = \frac{N_m^i}{N_m} \quad (3.4)$$

where p_m^i is the probability of estimation class i in the m th node. Another impurity measure is the Gini index, defined as

$$GI = \sum_{i=1}^K p_m^i (1 - p_m^i) \quad (3.5)$$

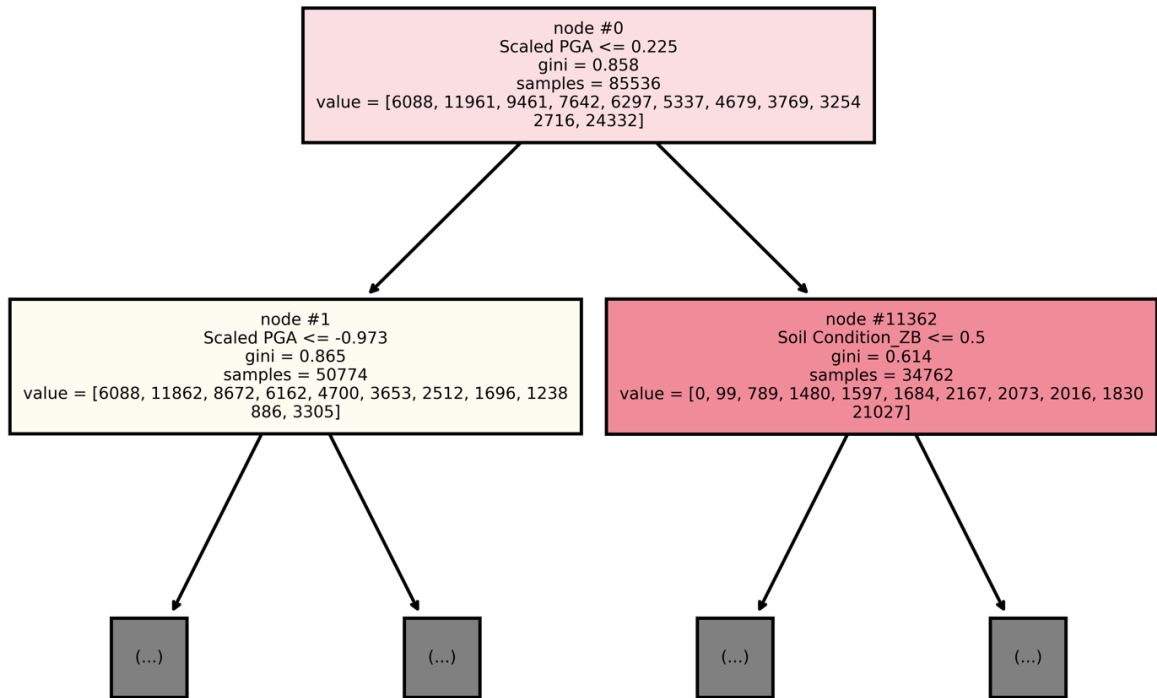


Figure 3.1. Two layers of the fitted best decision tree algorithm.

3.2.2. Random Forest

A random forest algorithm is an ensemble method of combining decision trees. The generalization error of a forest of tree classifiers is determined by the capacity of the individual trees in the forest as well as their correlation with each other. Due to the Law of Large Numbers, random forests are very efficient algorithms for prediction. The random forest algorithm further divides the training set into multiple subsets and implements the decision tree algorithm to each subset. The final result is provided by averaging the outputs of all decision trees. This process yields a remarkable enhancement in overall accuracy and control over-fitting (Breiman, 2001).

3.2.3. K-Nearest Neighbor

K-nearest neighbors (KNN) is a non-parametric machine learning algorithm that is based on a distance metric to identify the K nearest training samples to the new

instance for classification. The number of neighbors, K , is a hyperparameter that determines the amount of data used for classification. The class of the new instance is then assigned based on the majority class of its K nearest neighbors. Thus, the algorithm is based on the distance metric and the number of neighbors to make predictions. Figure 3.2 illustrates how the KNN algorithm classifies the output according to features with a simple example of the result of the KNN classification algorithm with trained two features.

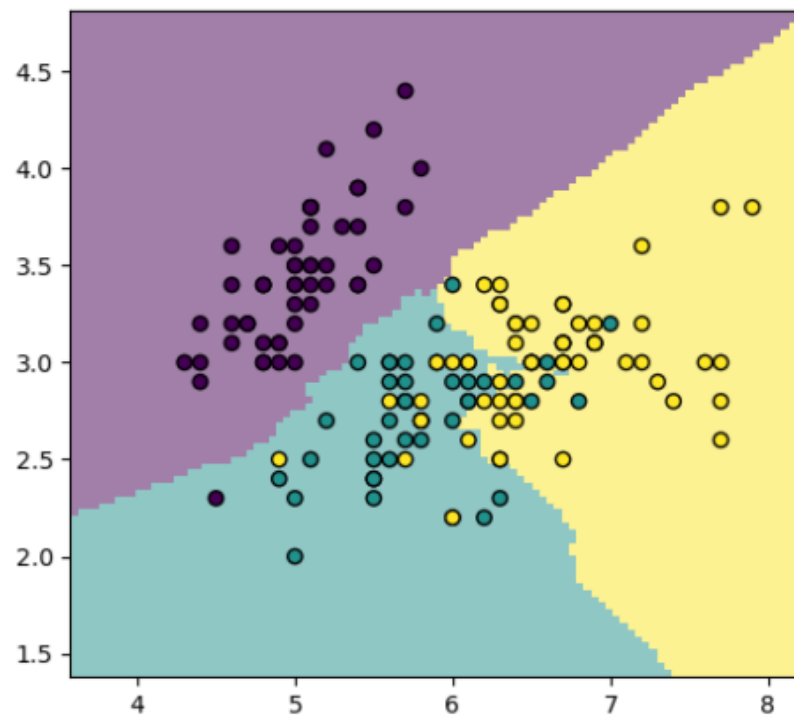


Figure 3.2. The result of an example of the KNN classification algorithm (Pedregosa et al., 2011).

3.2.4. Naïve Bayes

Unlike the other learning algorithms, the Naïve Bayes algorithm is not based on finding best-fit models of the data. The main idea of the Naïve Bayes algorithm is to estimate the most probable class according to a posterior probability distribution over models. The Naïve Bayes algorithm is based on Bayes' theorem, which is a probabilistic framework for combining prior information with observed data to make

inferences. By utilizing this framework, Naïve Bayes can obtain fast and accurate predictions for various classification problems (Do, 2017). The conditional probability can be calculated as:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (3.6)$$

where $P(C_i|X)$ is posterior, $P(X|C_i)$ is likelihood, $P(C_i)$ is prior probability, and $P(X)$ is marginal likelihood (scaling factor).

3.2.5. Stochastic Gradient Boosting

Stochastic Gradient Boosting (SGB) is an ensemble method that can be used for both regression and classification tasks. SGB combines multiple decision trees to make predictions. The algorithm works by fitting a decision tree to the data and then making predictions based on the residuals (i.e., the difference between the predicted and actual values) of the previous iteration. Breiman (1996) claims that adding randomness to a model improves the performance of the model. The stochastic Gradient Boosting model randomly selects sub-training data from complete data and fits the pseudo-residuals, which is a base learner, and it is calculated for each iteration (Freidman, 1999). The algorithm's steps are explained by Freidman (1999). Equations 25, 26, 27, 28, and 29 explain the steps of the algorithm.

$$F^*(x) = \underset{\gamma}{\operatorname{argmin}} \left(\sum_{i=1}^n L(y_i, \gamma) \right) \quad (3.7)$$

where $L(y_i, \gamma)$ is the loss function; in this thesis, the “log-loss” loss function is used for minimizing to predict better. The “log-loss” function can be defined as:

$$L = -(y_i \times \log(p) + (1 + y_i) \times \log(1 - p)) \quad (3.8)$$

where p is the predicted probability.

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \dots, n \quad (3.9)$$

where m is the decision tree for each iteration time while i is the single sample. r_{im} is residual. Then, a tree is trained with features x against residuals, and terminal nodes

of the three are created. Terminal nodes are represented by j in the equation below. Moreover, the loss function minimizes each terminal node by calculating γ_{jm} .

$$\gamma_{jm} = \operatorname{argmin} \left(\sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma) \right) \text{ for } j = 1, \dots, J_m \quad (3.10)$$

where γ_{jm} minimizes the “log-loss” function on each terminal node. The last step is updating the model with the predicted output, in which the log-loss function is minimized.

$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} 1(x \in R_{jm}) \quad (3.11)$$

In Equation 3.11, v is the learning rate that controls model over-fitting.

3.3. Metrics of Evaluating the Model Performance

Each ML algorithm used takes a distinct approach to data interpretation. Finding how well a model fits the data is a crucial element of the ML process. Metrics assess the model’s performance. There are numerous methods for evaluating categorization models. Some of them (the confusion matrix, F_1 score, accuracy, precision, and recall) are used in the thesis to identify which algorithms are the best fit for the data.

3.3.1. Accuracy

Accuracy is the most basic metric, defined as the number of correct predictions divided by the total number of forecasts.

$$\text{Accuracy} = \frac{\text{Correct Prediction}}{\text{Total Number of Prediction}} \quad (3.12)$$

3.3.2. Confusion matrix

As shown in Table 3.2, the confusion matrix is a tabular representation of the actual class versus model expectations. While diagonal elements represent true predictions, off-diagonal elements represent false predictions.

Table 3.2. Confusion matrix.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

3.3.3. Precision and Recall

When data is imbalanced due to a class's dominance, accuracy is not a sufficient indicator of the model. Not only should the most common classes be correctly predicted, but so should the rare classes. Precision is the number of instances correctly placed in the positive anticipated class, whereas recall is the ratio of correct positive predictions to the positive actual class. Equations 3.13 and 3.14 define these, respectively.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \quad (3.13)$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (3.14)$$

3.3.4. F_1 Score

The F_1 score is calculated by combining recall and precision. These two criteria are crucial in evaluating the model. As a result, precision and recall are merged to form a single significant metric. It should be noted that precision and recall are not mutually exclusive. If one of them rises, the other falls. Nonetheless, the greater the F_1 score, the better the model's performance.

$$F_1score = \frac{2 \times precision \times recall}{precision + recall} \quad (3.15)$$

3.4. Data Processing for Machine Learning Algorithms

This section includes explaining the raw data, visualizing the raw data to express the distribution of the data in terms of output, and the data pre-processing to prepare data for implementation of machine learning algorithms. Data pre-processing consists of 5 different processes. These are data representation, data standardization, data normalization, data discretization, and feature selection. These processes are mentioned in Section 3.4.2.

3.4.1. Raw Data

The raw data consists of 106,920 analysis results with building characteristics, soil condition, intensity measures, and MIDR. Table 3.3 shows the features of raw data, while Table 3.4 shows the output of the dataset. Machine learning models are trained to utilize each IM separately. MIDR is divided into 10 discrete values between 0 and 0.01.

Table 3.3. Features and data types of raw data.

Feature Names	Data Type
Number of Stories	Numerical
Number of Spans	Categorical
Span Length	Numerical
Story Height	Numerical
Column Width	Numerical
Column Depth	Numerical
Concrete Strength	Numerical
Commercial Use	Categorical
Soil Condition	Categorical
PGA (g)	Numerical
PGV (cm/s)	Numerical
PGD (cm)	Numerical
$S_a(T_1)$ (g)	Numerical
$S_d(T_1)$ (m)	Numerical
CAV (m/s)	Numerical

Table 3.4. Target (output) of the dataset.

Feature Names	Data Type
MIDR Level	Categorical

The conditional distribution of MIDR conditioned on each input is given in the figures below to highlight the relationship between inputs and outputs. The distribution of MIDR conditioned on column depth for each number of stories is illustrated in Figure 3.3 and Figure 3.4. The reason for showing distribution with respect to the number of stories is that column width is related to the number of stories. These distributions show that as column depth or column width decreases, MIDR increases.

Columns are assumed to be square; therefore, the distribution of MIDR conditioned on column depth and column width is the same. The values illustrated in the legends are cm for column depth and span length.

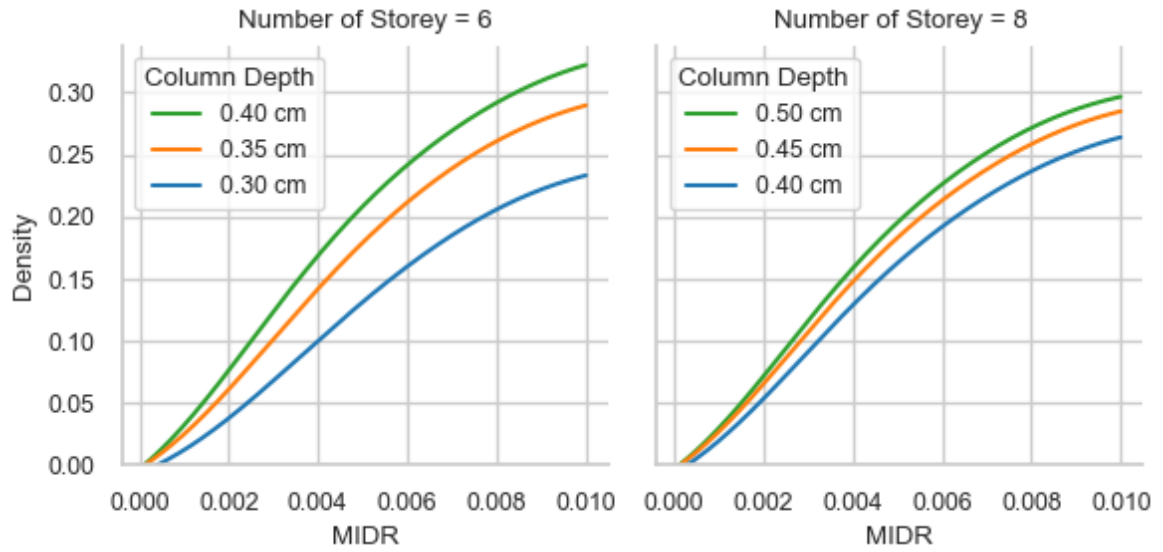


Figure 3.3. The conditional cumulative distribution function of MIDR conditioned on column depth.

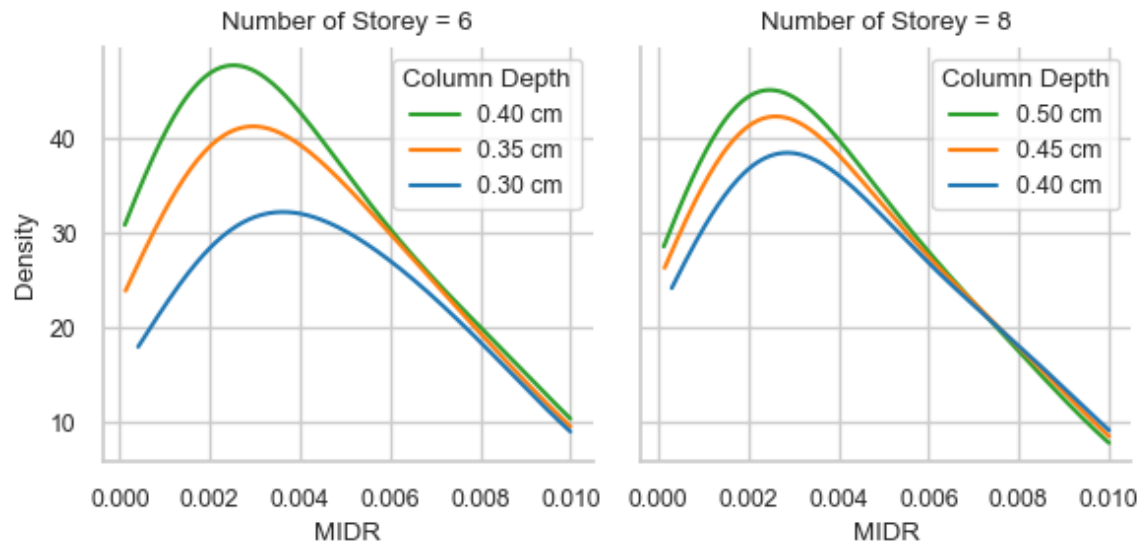


Figure 3.4. The conditional probability density function of MIDR conditioned on column depth for 6 and 8-story.

Another substantial conditional probability is MIDR conditioned on the story height. Figure 3.5 shows the right-skewed conditional probability density function of MIDR. As can be seen there, if story height increases, rigidity decreases. Thus, the non-exceedance probability of MIDR is conditioned on 2.6 m, which is higher than 3.0 m for low MIDRs.

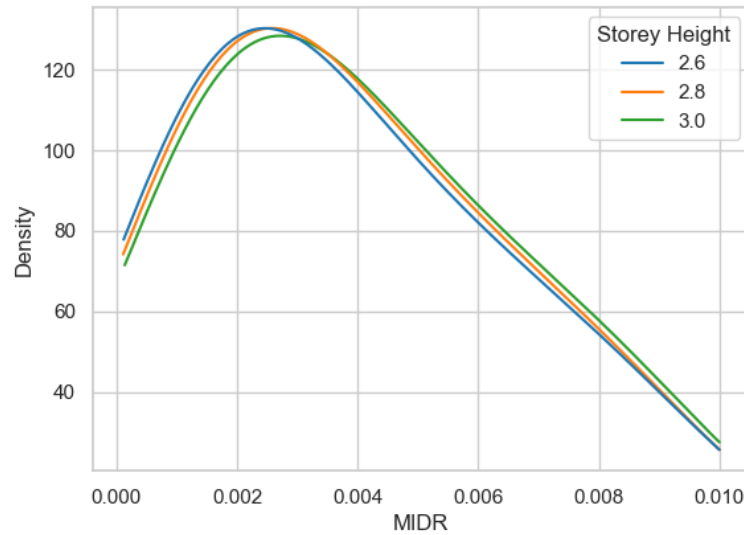


Figure 3.5. The conditional probability density function of MIDR conditioned on story height.

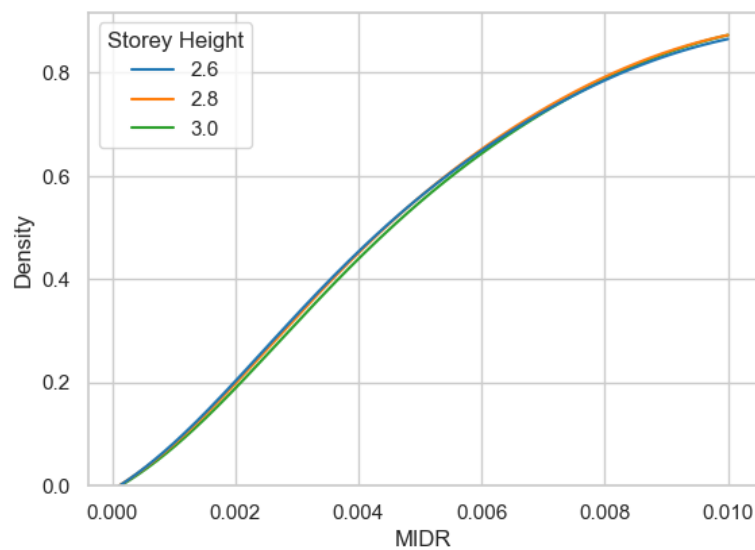


Figure 3.6. The conditional cumulative distribution function of MIDR conditioned on story height.

Moreover, span length is also examined to reveal how it affects the MIDR. Figure 3.7 and Figure 3.8 demonstrate the probability density function and cumulative distribution function of MIDR conditioned on span length, respectively. These figures emphasize that higher span lengths cause higher MIDRs than lower span lengths. The reason for this result is the same as with column dimension or story height. When span length increases, which means beam length increases, the rigidity of the buildings decreases. So, when a building having this issue faces strong ground motion, it is more likely to have more damage than a more rigid building.

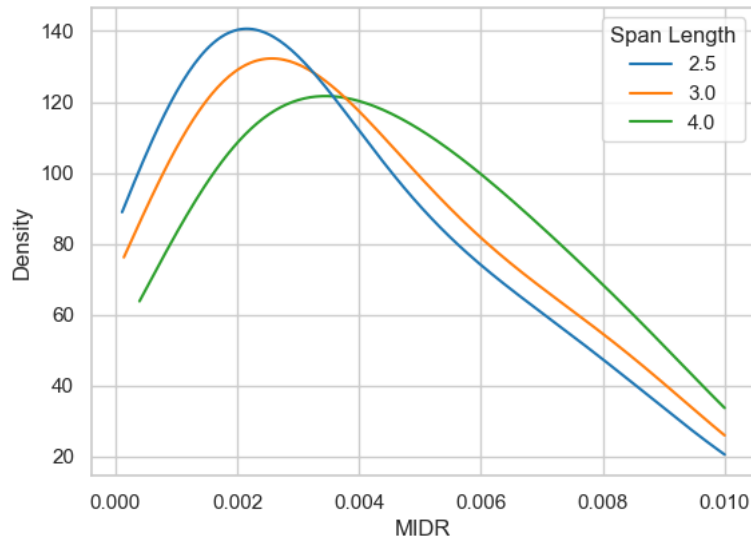


Figure 3.7. The conditional probability density function of MIDR conditioned on span length.

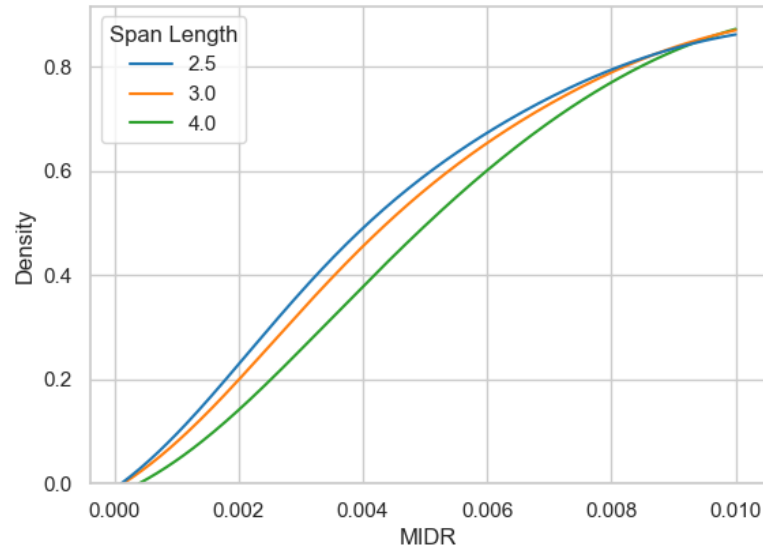


Figure 3.8. The conditional cumulative distribution function of MIDR conditioned on span length.

Another variation is considered in the concrete compression strength of buildings. Figure 3.9 and Figure 3.10 illustrate the MIDR distribution conditioned to concrete compression strength. As expected, buildings with low concrete strength tend to suffer more damage under strong ground motion.

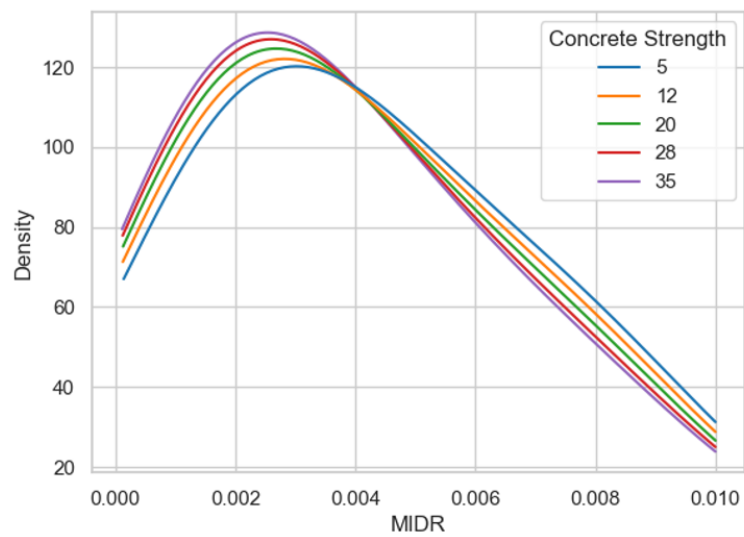


Figure 3.9. The conditional probability density function of MIDR conditioned on concrete strength.

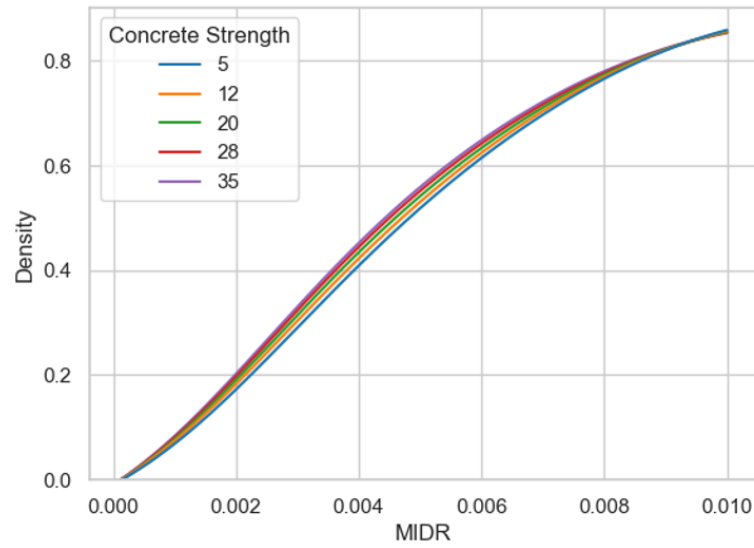


Figure 3.10. The conditional cumulative distribution function of MIDR conditioned on concrete strength.

Figures 3.11, 3.13, and 3.15 demonstrate the probability density function of MIDR conditioned on commercial use of the first story, soil condition type, and period class, respectively. The cumulative distribution functions are shown in Figures 3.12, 3.14, and 3.16.

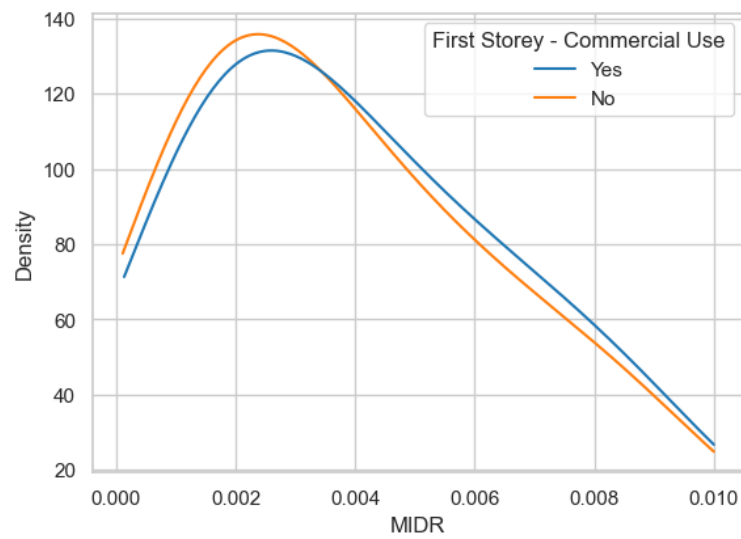


Figure 3.11. The conditional probability density function of MIDR conditioned on commercial use of the first story.

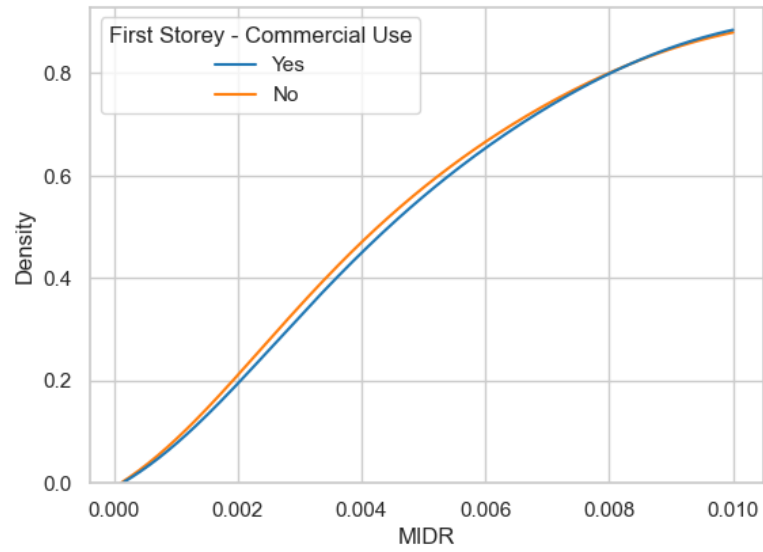


Figure 3.12. The conditional cumulative distribution function of MIDR conditioned on commercial use of the first story.

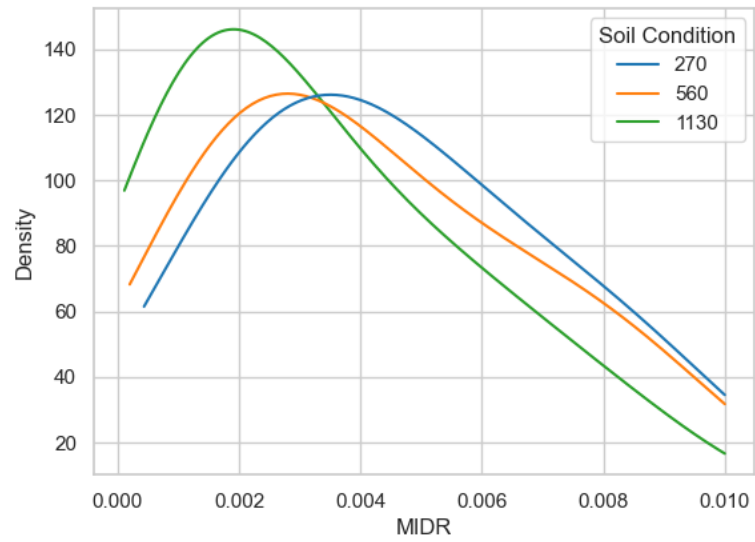


Figure 3.13. The conditional probability density function of MIDR conditioned on soil conditions.

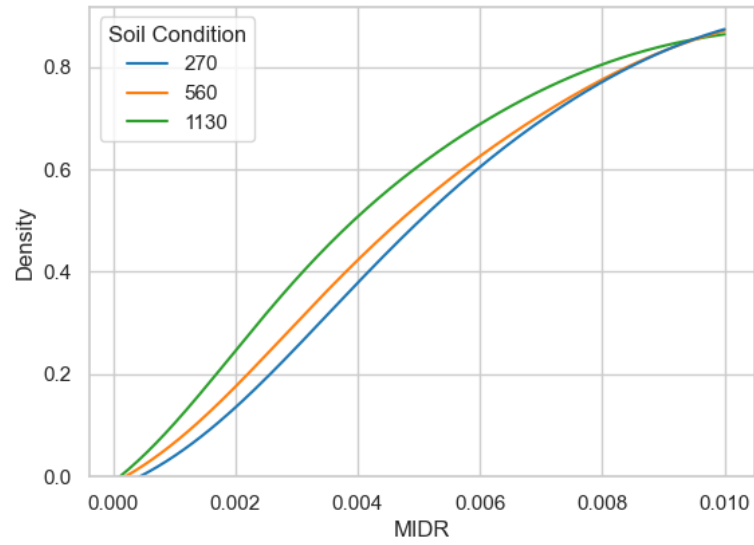


Figure 3.14. The conditional probability density function of MIDR conditioned on soil conditions.

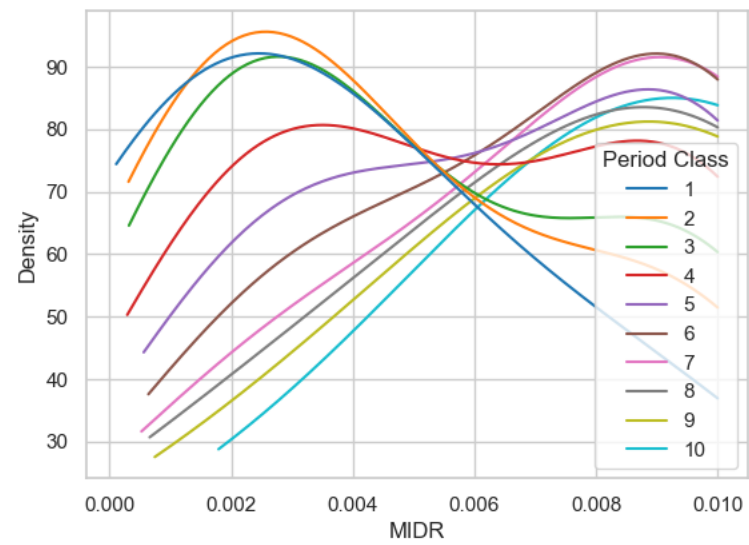


Figure 3.15. The conditional probability density function of MIDR conditioned on period class.

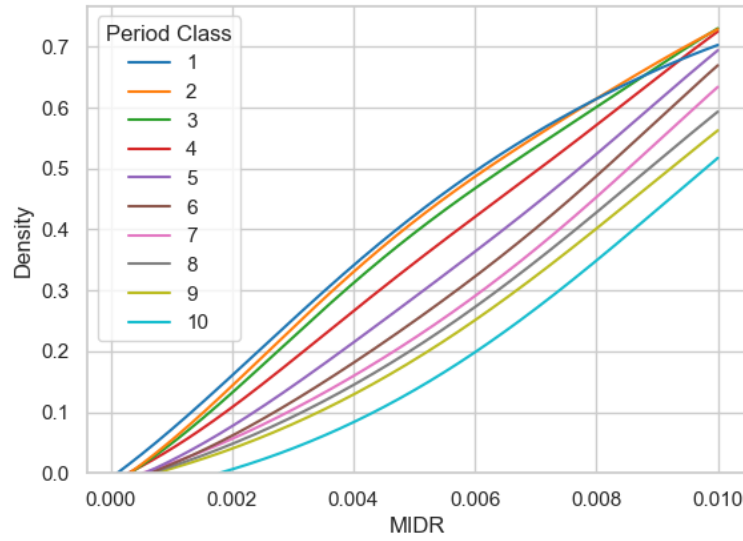


Figure 3.16. The conditional cumulative distribution function of MIDR conditioned on period class.

MIDR-IMs scatter diagrams are presented in Figures 3.17, 3.18, 3.19, 3.20, 3.21, and 3.22. These figures illustrate the MIDR distribution for each IM used in the scope of this thesis. The results also indicate that when the IM values increase, so does the MIDR. These figures also contain the marginal distribution of MIDR at the right-hand side of the scatter plots and the marginal distribution of IMs at the top of the scatter plots. These marginal distributions show how IMs and MIDR are distributed independently.

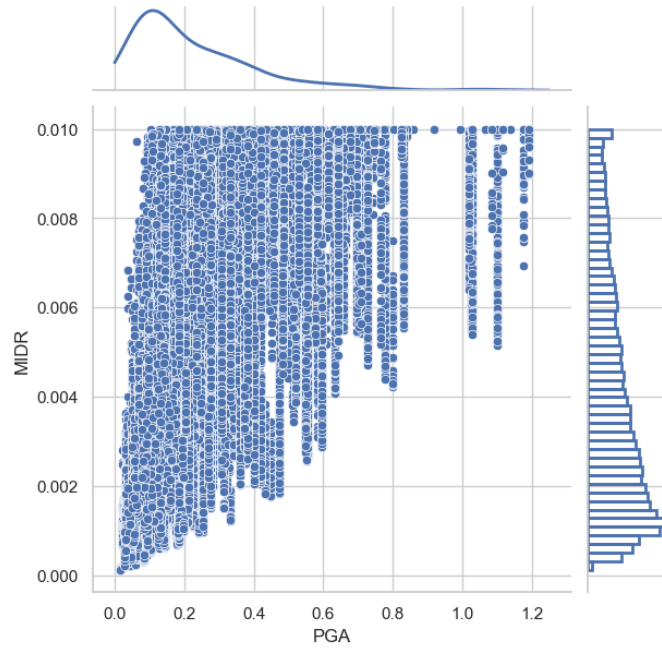


Figure 3.17. Scatter plot of MIDR vs. PGA for the representative buildings with the marginal distributions of MIDR and PGA.

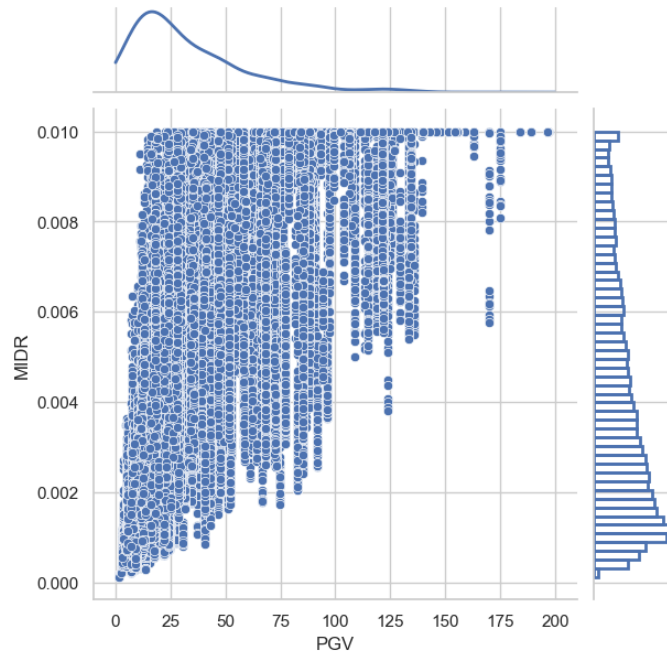


Figure 3.18. 40. Scatter plot of MIDR vs. PGV for the representative buildings with the marginal distributions of MIDR and PGV.

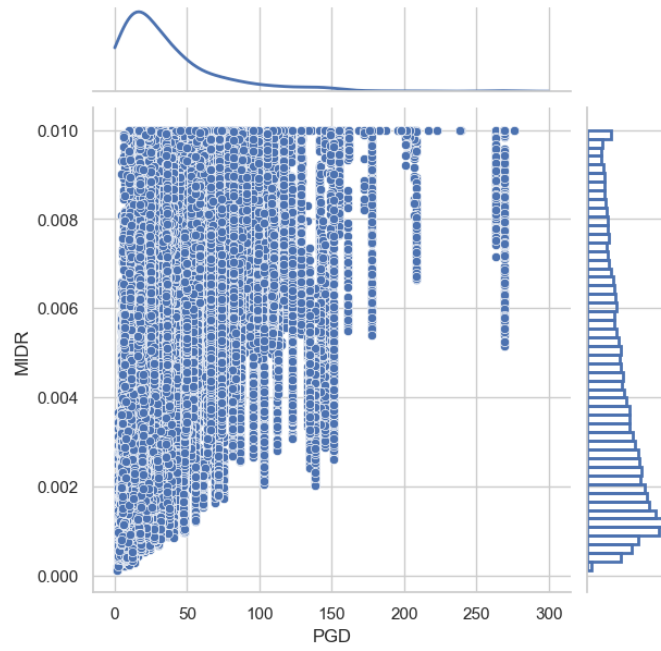


Figure 3.19. Scatter plot of MIDR vs. PGD for the representative buildings with the marginal distributions of MIDR and PGD.

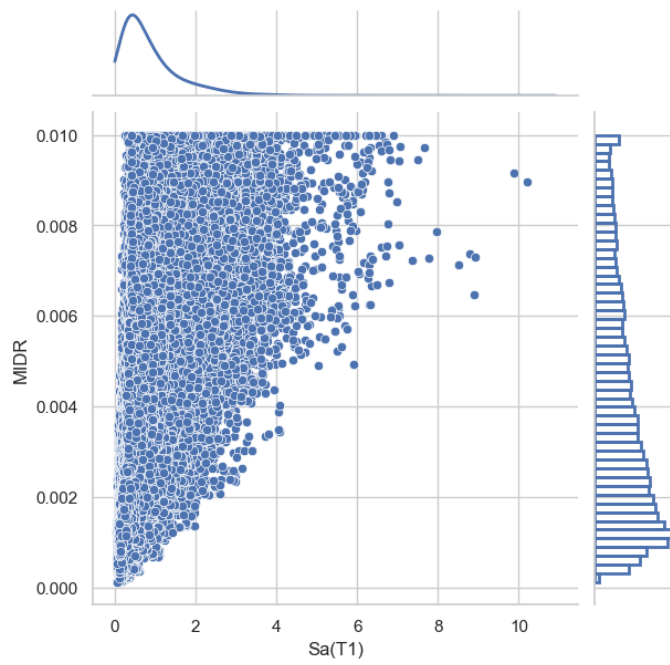


Figure 3.20. Scatter plot of MIDR vs. $S_a(T_1)$ for the representative buildings with the marginal distributions of MIDR and $S_a(T_1)$.

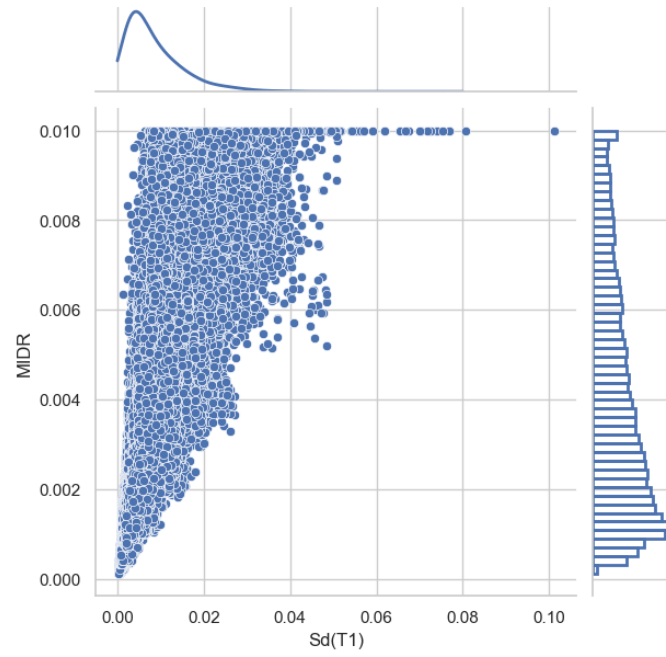


Figure 3.21. Scatter plot of MIDR vs. $S_d(T_1)$ for the representative buildings with the marginal distributions of MIDR and $S_d(T_1)$.

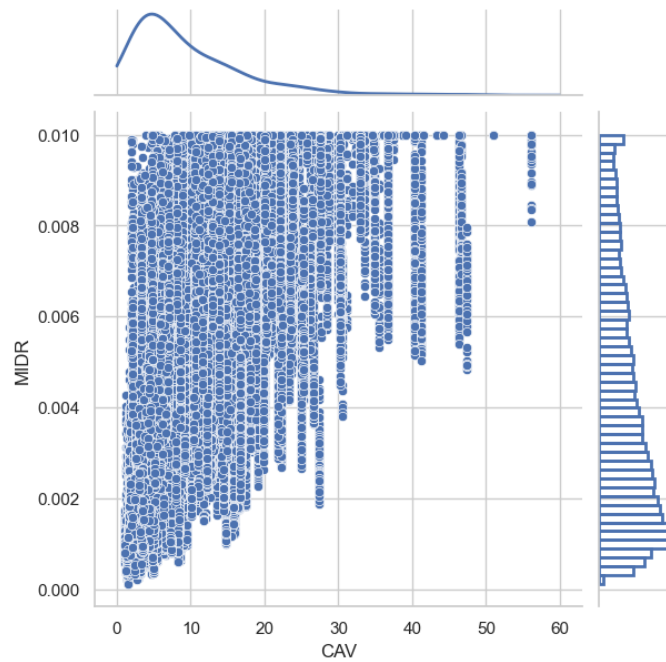


Figure 3.22. Scatter plot of MIDR vs. CAV for the representative buildings with the marginal distributions of MIDR and CAV.

Figure 3.23 represents the scatter diagrams of each IM against the others. The figure shows that as one IM increases, the other IMs also increase, but this increase is partially positively correlated rather than fully correlated. This is evident from the dispersion of IMs plotted against each IM in the graphs.

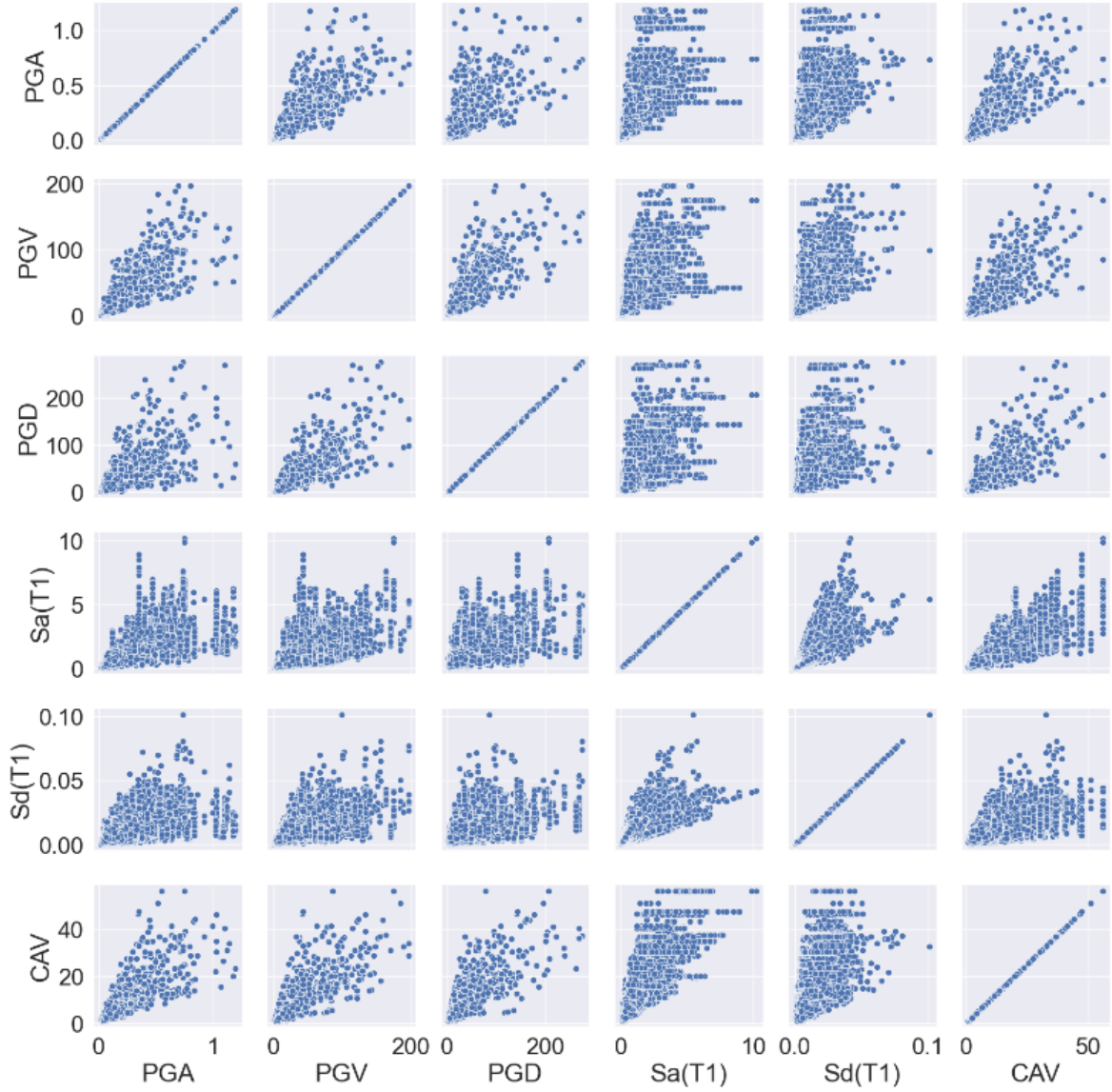


Figure 3.23. Scatter plot of each IM against the others.

3.4.2. Pre-Processing of Raw Data

The dataset is artificially created; thus, there are no missing values or erroneously entered values. The first step of preprocessing data would be the imputation of these missing values according to the appropriate method after deleting erroneously entered data. There are several methods, such as filling missing values with the mean of the feature, random values conditioned on other features, or dropping the row. However, as mentioned before, the dataset is synthetic, so this step was not applied to the dataset.

The second step in preparing the dataset for machine learning is the detection of outliers. Intensity measures can take impossible values, such as exceeding 200 cm/s in PGV, due to utilizing conditional mean spectrum and multiplying the record with a scale factor. Therefore, primarily, outliers in IMs were deleted from the dataset to misinterpret the outcome of ML algorithms. These outliers of IMs considered in this thesis are given in Table 3.5.

Table 3.5. Detection of Outliers of IMs.

Intensity Measures	Restriction
PGV	≤ 200 cm/s
PGA	≤ 1.5 g
PGD	≤ 300 cm
CAV	≤ 60 m/s

Outliers are calculated by using the interquartile range (IQR) method. First of all, the 25th and 75th percentiles of the data are determined. Then, the interquartile range between these percentiles is calculated by subtracting them from each other. The final step is controlling whisker length. The whisker lengths are selected as 3.2, 2.25, 3.95, and 2.75 for PGA, PGV, PGD, and CAV, respectively. Moreover, the other IM parameters are shown in the boxplot with a whisker length of 1.5. However, these parameters are not restricted by interquartile range because after applying restrictions

to PGA, PGV, PGD, and CAV, the other IMs are subsequently considered outliers. Figure 3.24 demonstrates the interquartile range of each IM with a boxplot while Figure 3.25 demonstrates the boxplot (1.5 IQR) after deleting outliers from the data.

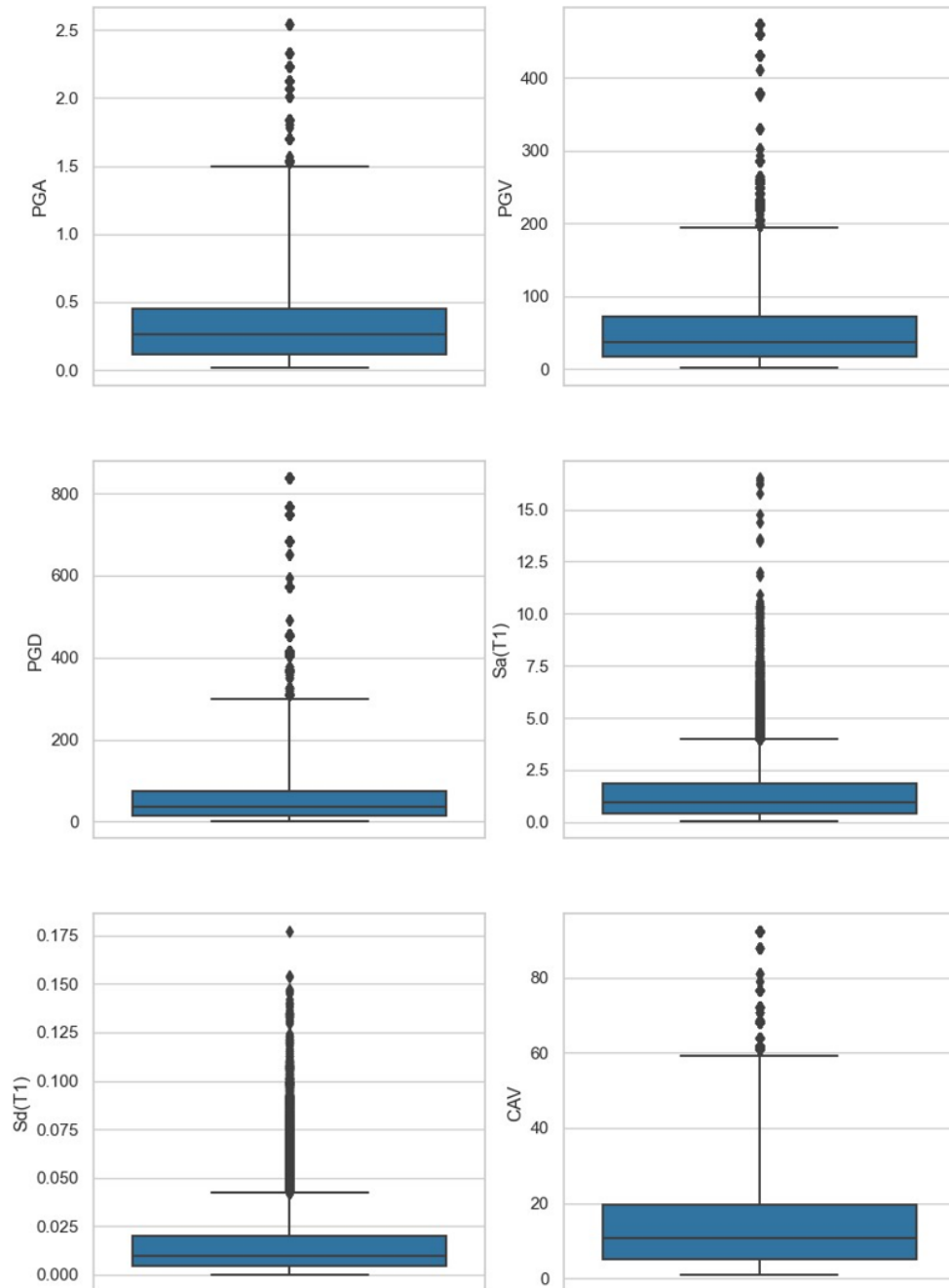


Figure 3.24. Boxplot to detect the outliers of IMs. [25-75 percentiles of the data (blue box), interquartile range (the outermost lines), the median values of the data (the lines between these percentiles), the outliers (the rhombus shapes)]

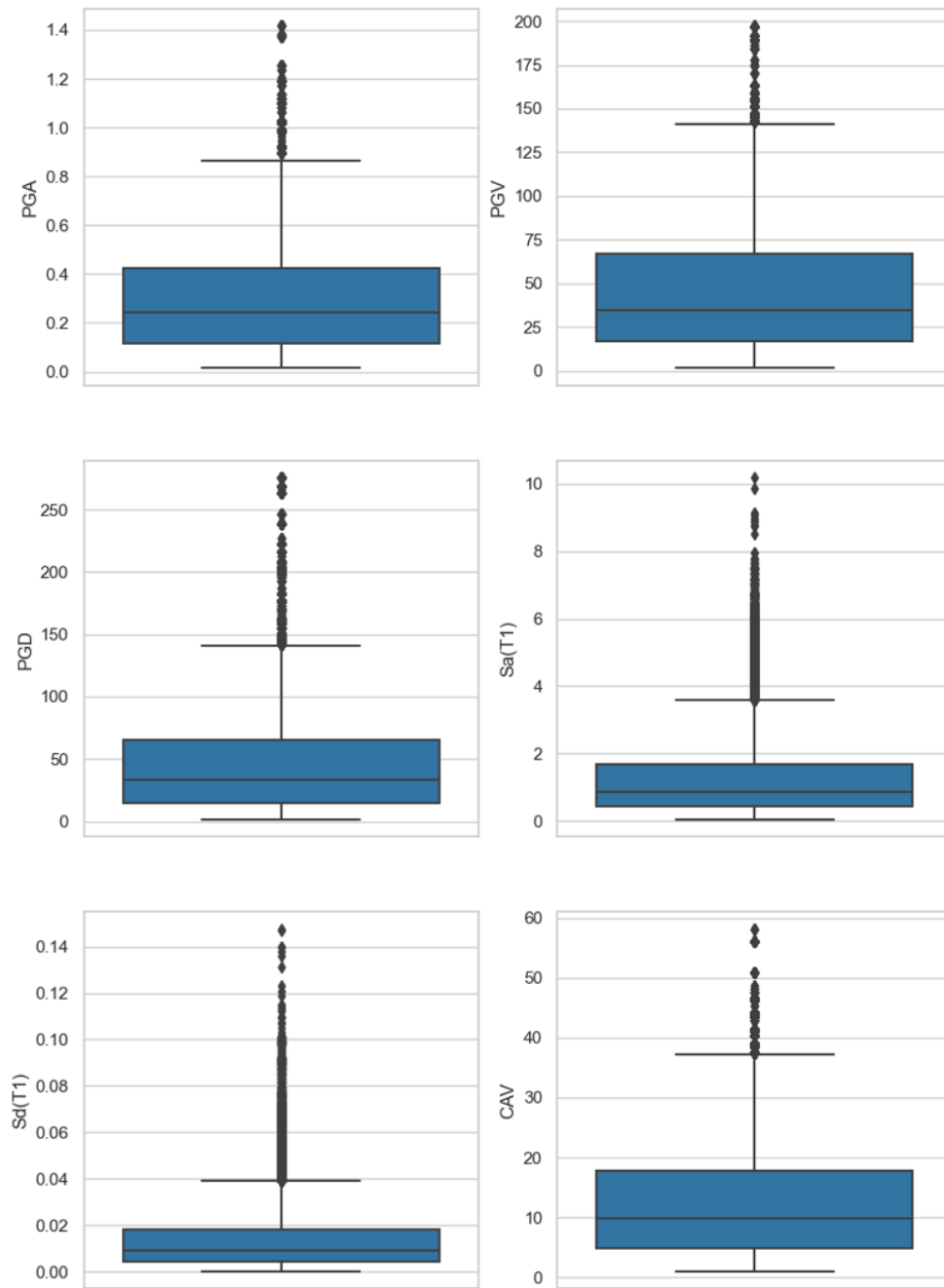


Figure 3.25. Boxplot (1.5 IQR) after deleting outliers from the data.[25-75 percentiles of the data (blue box), interquartile range (the outermost lines), the median values of the data (the lines between these percentiles), the outliers (the rhombus shapes)]

The blue boxes in Figure 3.24 and 3.25 indicate 25-75 percentiles of the data while the lines between these percentiles indicate the median values of the data. Moreover, the area between the outermost lines shows the interquartile range, and the rhombus shapes are defined as the outliers.

The third step of preprocessing data is the feature scaling. Feature scaling methods can be applied to the numerical features of the dataset. There are numerous feature scaling methods. One of the widely known methods is standardization. Standardization is also called Z-score normalization. Z-score transformation helps to rescale the distribution of each feature with a mean of 0 and a standard deviation of 1. This is also called standard normal distribution. Equation 3.16 shows how to calculate the Z-score.

$$z = \frac{x - \mu}{\sigma} \quad (3.16)$$

where x is a random variable, μ is the mean of the population, and σ is the standard deviation of the population.

Another feature scaling method is scaling features to a range. This scaling can involve either scaling data between the minimum and maximum values, distributing the data between 0 and 1, or scaling by dividing all data by the absolute maximum value on a feature; in this case, the data will be distributed between -1 and 1. Additionally, another feature scaling method is employed for the considered dataset. Non-linear transformation was used to transform the distribution of IMs to the standard normal distribution. Two types of non-linear transformations help to map actual distribution to a standard normal distribution. These are Power Transform and Quantile Transform. Power transforms are a parametric, monotonic transformation of data by utilizing power functions. Scikit-learn provides two different power transformations, which are Yeo and Johnson's (2000) and Box and Cox's (1964) transformers. Yeo-Johnson transformation works with positive and negative values, while Box-Cox transformation only works with positive values. Quantile transforms firstly estimate the cumulative

distribution function of a feature, then the obtained values are mapped to the standard normal distribution by utilizing the quantile function.

The other significant transformation is transforming log-normal distribution to normal distribution. As mentioned before, the distributions of story height, column width, and column depth features are log-normally distributed. The logarithm of the random variable of the log-normal distribution is normally distributed. Therefore, the mean value and standard deviation of the normal distribution can be calculated with Equations 3.17 and 3.18, respectively. Then, the Z-score transformation can be applied to the normal distribution to obtain a standard normal distribution.

$$\mu = \ln \left(\frac{\mu_x^2}{\sqrt{\mu_x^2 + \sigma_x^2}} \right) \quad (3.17)$$

where μ is the mean of the variable's natural logarithm, μ_x is the desired mean, and σ_x is the desired standard deviation.

$$\sigma = \sqrt{\ln \left(1 + \frac{\sigma_x^2}{\mu_x^2} \right)} \quad (3.18)$$

where σ is the standard deviation of the variable's natural logarithm. The Z-score transformation can be written as:

$$z = \frac{\ln(x) - \mu}{\sigma} \quad (3.19)$$

The feature of story height is log-normally distributed. This distribution, shown in Figure 3.26, is converted to the standard normal distribution to train ML algorithms.

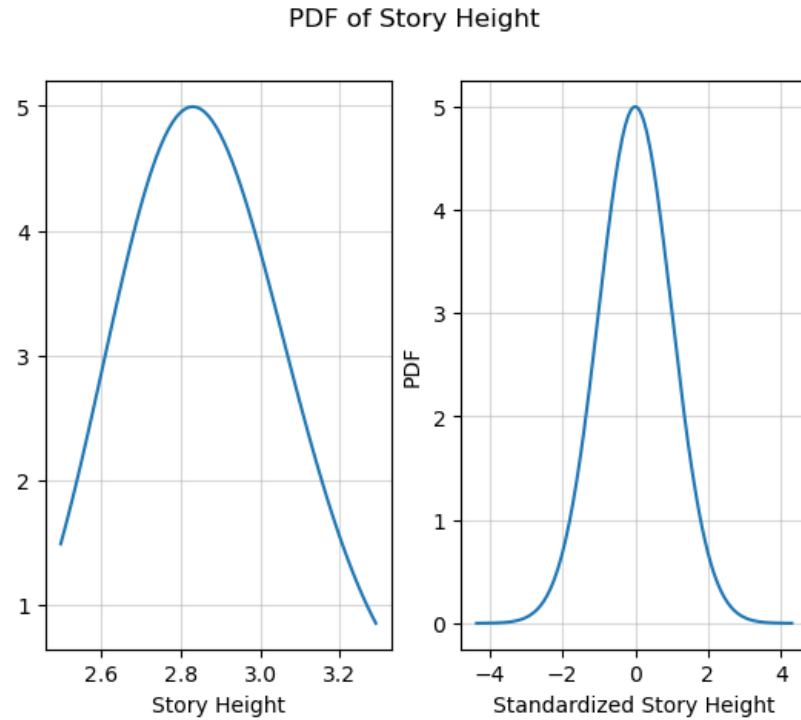


Figure 3.26. The probability density function of story height. (left) Actual distribution of story height, (right) standardized distribution of story height

The distribution of the span length feature is gamma distribution with a mean of 3.37 and a coefficient of variation of 0.38. The actual distribution and standardized distribution of span length are given in Figure 3.27.

Figure 3.28 demonstrates the actual and standardized distribution of concrete compression strength feature, which is gamma distribution with a mean of 16.73 and a coefficient of variation of 0.51.

Column depth and column width features are normally distributed conditioned on the number of stories. When the number of stories is 6, the mean and standard deviation of the distribution of column width and depth are 0.35 and 0.05, respectively. Additionally, column width and depth are normally distributed with a mean of 0.45 and a standard deviation of 0.05, when the number of stories is 8. Z score standardization is applied to these features.

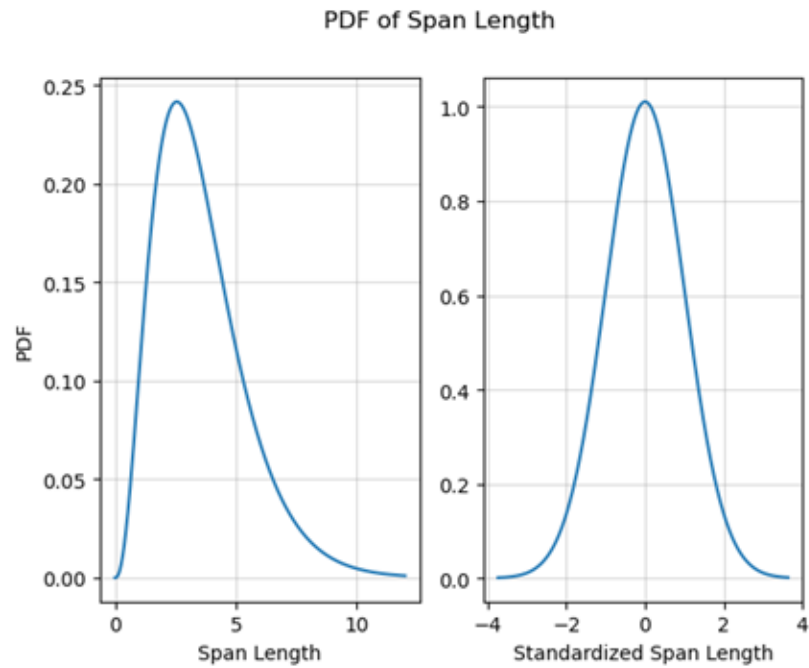


Figure 3.27. Probability density function of span length. (left) Actual distribution of span length, (right) standardized distribution of span length

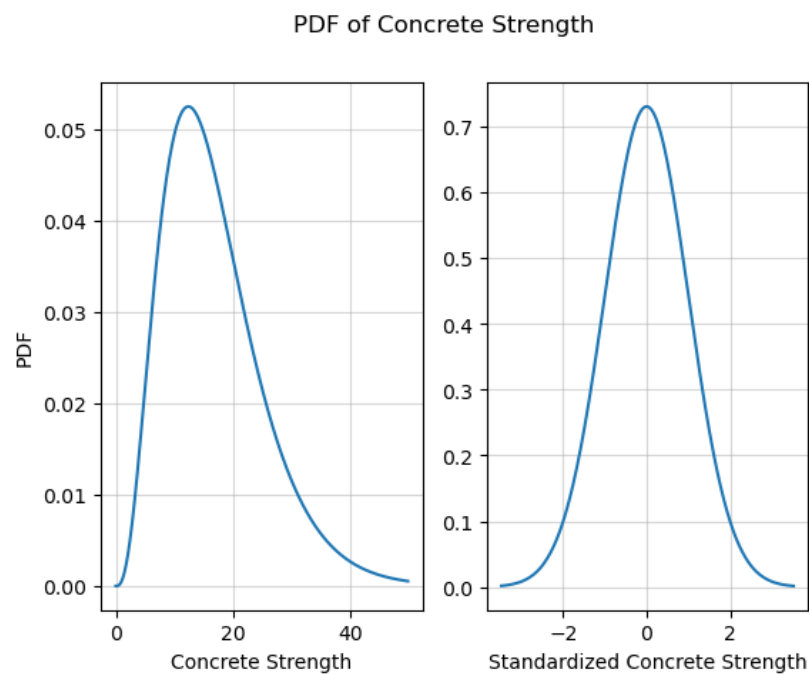


Figure 3.28. The probability density function of concrete compression strength. (left) Actual distribution of concrete compression strength, (right) standardized distribution of concrete compression strength

The distribution of PGA is transformed to a standard normal distribution by utilizing a Quantile Transformer, while other distributions of IM parameters are transformed to a standard normal distribution by utilizing a Power Transformer with the Box-Cox method. Figures 3.29, 3.30, 3.31, 3.32, 3.33, and 3.34 illustrate the actual and standard normal distributions of PGA, PGV, PGD, $S_a(T_1)$, $S_d(T_1)$, and CAV, respectively.

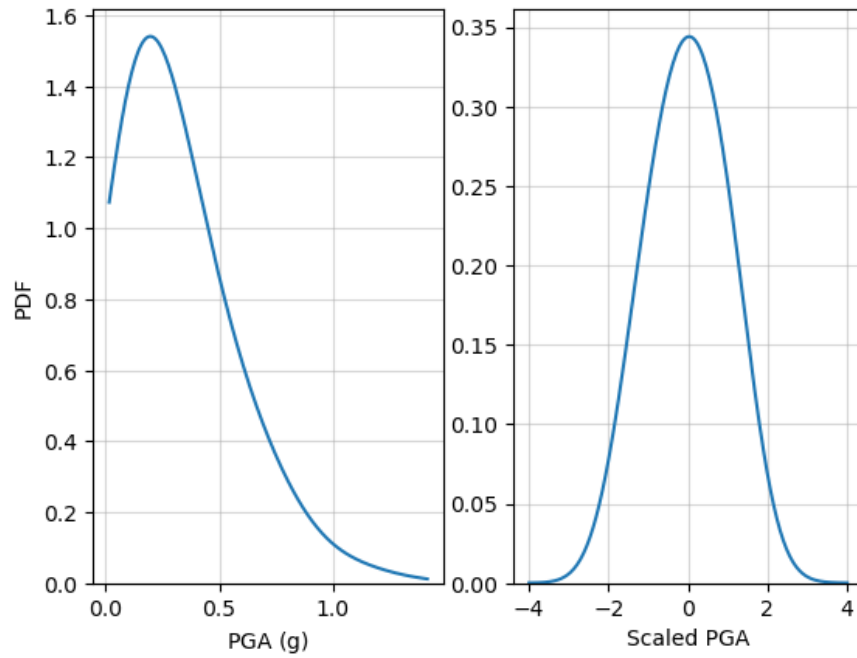


Figure 3.29. The probability density function of PGA (g). (left) Actual distribution of PGA, (right) standardized distribution of PGA

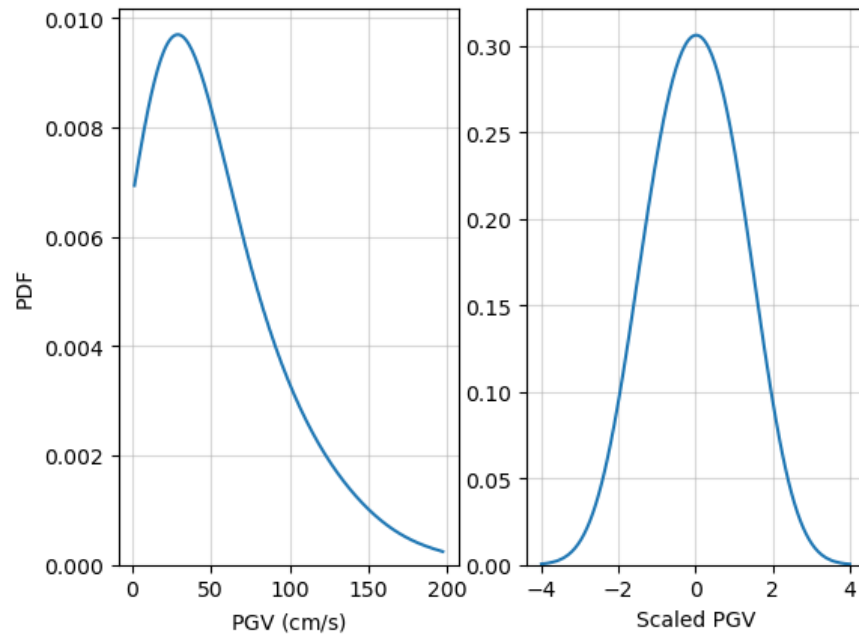


Figure 3.30. The probability density function of PGV (cm/s). (left) Actual distribution of PGV, (right) standardized distribution of PGV

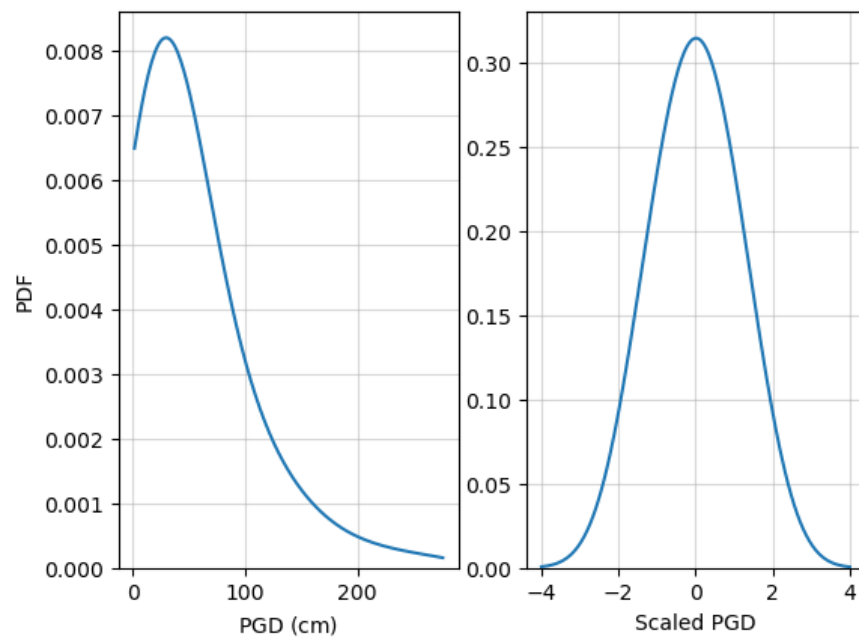


Figure 3.31. The probability density function of PGD (cm). (left) Actual distribution of PGD, (right) standardized distribution of PGD

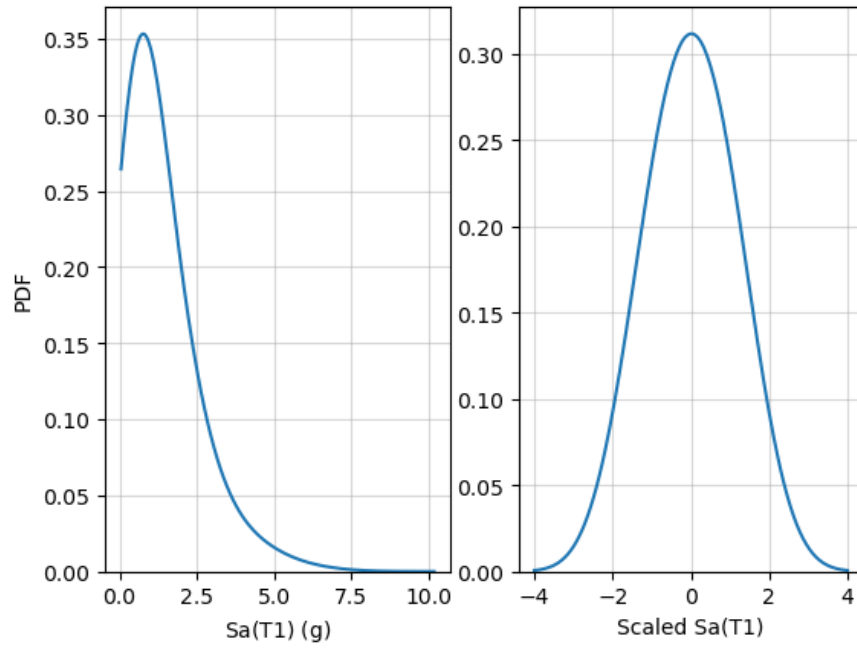


Figure 3.32. Probability density function of $S_a(T_1)$ (g). (left) Actual distribution of $S_a(T_1)$, (right) standardized distribution of $S_a(T_1)$

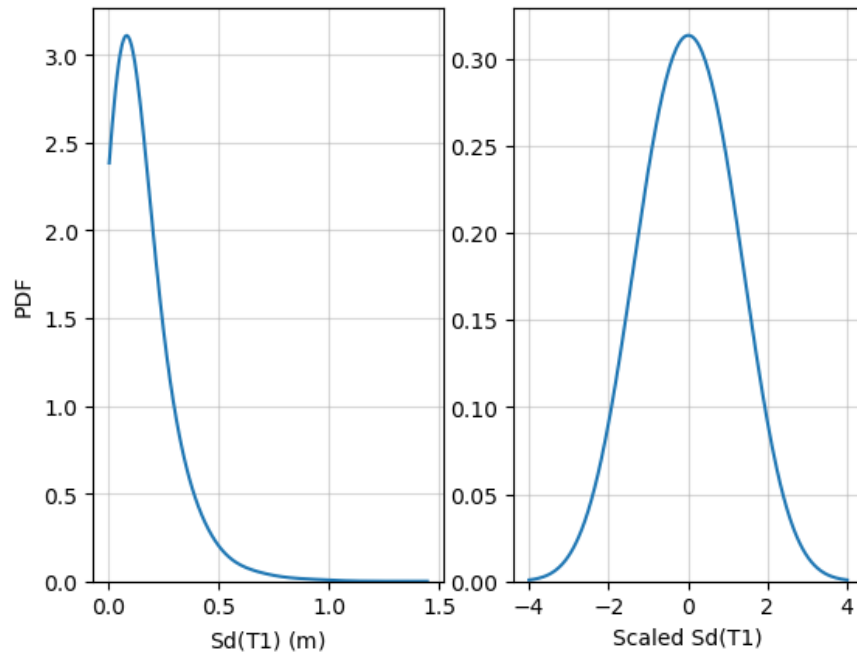


Figure 3.33. Probability density function of $S_d(T_1)$ (g). (left) Actual distribution of $S_d(T_1)$, (right) standardized distribution of $S_d(T_1)$

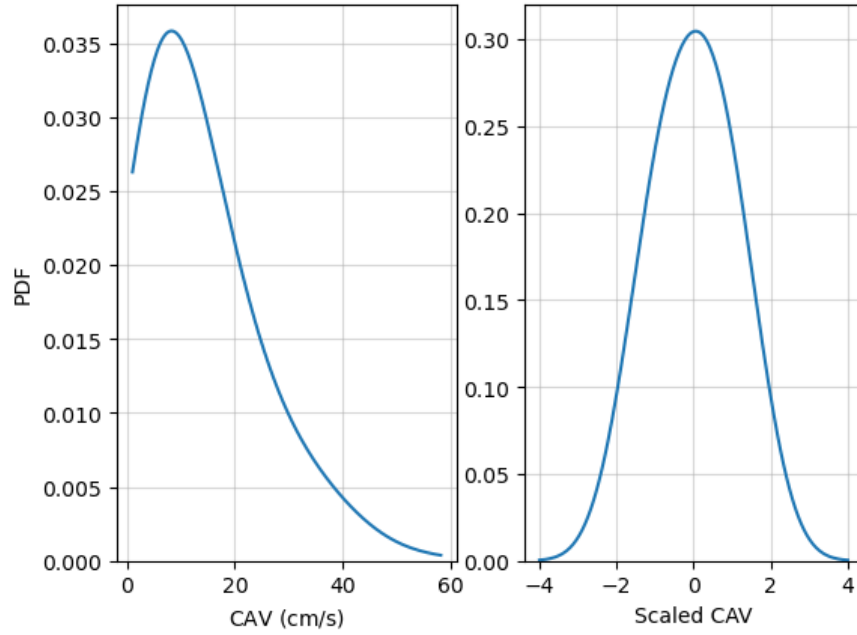


Figure 3.34. The probability density function of CAV (m/s). (left) Actual distribution of CAV, (right) standardized distribution of CAV

Another preprocessing method applied to the dataset is min-max scaling to the feature of the number of stories.

The last step of preprocessing the dataset is handling with categorical features. This type of data should be converted to binary vectors. The process of binarization of features is called the encoding process. To give an example of encoding, commercial use of the first story is categorical data and encoded by applying the one-hot encoding method to features and obtain Table 3.6.

Table 3.6. Example of one-hot encoding.

Data	Commercial Use_No	Commercial Use_Yes
1	0	1
2	0	1
3	1	0
4	1	0

The training dataset is prepared to establish machine learning algorithms after preprocessing methods are applied to the data. An example of the final version of the dataset is illustrated in Table 3.7. The total amount of data in the dataset after preprocessing operations is 101,855.

Table 3.7. An example of the final version of the dataset.

Data Label	Data 81998	Data 12687	Data 101305	Data 53954
DamageState	0	2	1	6
First Story–Commercial Use_No	0	0	1	0
First Story–Commercial Use_Yes	1	1	0	1
Soil Condition_ZB	1	1	0	0
Soil Condition_ZC	0	0	0	1
Soil Condition_ZD	0	0	1	0
Number of Spans_4Span	1	0	0	1
Number of Spans_5Span	0	1	1	0
Scaled Story Height	-1.065	-0.138	0.726	0.726
Scaled Span Length	-0.576	-0.576	-0.114	-0.576
Scaled Concrete Strength	1.184	1.184	1.593	-1.969
Scaled Column Width	1	0	0	1
Scaled Column Depth	1	0	0	1
Scaled Number of Stories	0.25	0.25	0.75	0.75
Scaled PGA	-1.358	0.759	-0.545	-0.044
Scaled PGV	-1.580	0.246	-0.750	0.166
Scaled PGD	-1.147	0.142	-1.344	1.273
Scaled $S_a(T_1)$	-2.157	0.052	-0.601	1.050
Scaled $S_d(T_1)$	-2.811	-0.467	-0.602	1.004
Scaled CAV	-1.322	0.654	-0.643	0.947

4. THE RESULT OF MACHINE LEARNING ALGORITHMS AND THE EVALUATION OF PERFORMANCE METRICS OF THE ALGORITHMS

The final dataset is split into two parts: the training and test sets. The training set is chosen by randomly selecting 80% of the dataset, which is 81,484 data points, whereas the rest of the data, which is 20,371 data points, is used as the test set. ML algorithms are trained by using the training set, while the test set is used to examine the performance of the ML algorithms. However, if ML algorithms are only trained with the whole training set, the over-fitting or under-fitting problem can emerge. Therefore, the training set is split into 10 cross-validation subsets with the Stratified K-Folds method. The primary purpose of this process is to prevent over-fitting. Over-fitting occurs when the model becomes too specific in its training stage on a certain dataset; it performs poorly on new, unseen data. This problem is characterized by the model gathering too many features and complexities from the training data, affecting its capacity to adapt to new data points. As a result, when confronted with new, unfamiliar data, the model's precision and potential suffer because it struggles to find essential trends within the breadth and complexity of the total data distribution.

Additionally, the hyperparameters of each ML algorithm are tuned to increase the prediction's performance. The hyperparameters for decision tree algorithms to be tuned are minimum sample split and maximum depth. The minimum sample split indicates the number of samples required to split an internal node, while the maximum depth indicates the depth of the tree (Pedregosa et al., 2011).

Random forest hyperparameters are also tuned. These hyperparameters are the maximum features (the number of features considered for the best split), the minimum sample splits (the minimum number of samples needed to separate an internal node), and the minimum sample leaf (the minimum number of samples necessary at a leaf node). A split point will be considered at any depth if it leaves at least a minimum value

of training samples in both the left and right branches and the number of estimators, which indicates the number of trees in the forest (Pedregosa et al., 2011).

Another hyperparameter tuning is done for the KNN algorithm. The hyperparameters of KNN algorithms tuned are the number of neighbors, weights, and metric. The meanings of these hyperparameters are the number of neighbors to use for classifying neighbors, weights of all points in each neighborhood, and distance metric, which is used while calculating the distance between predicted and actual, respectively. The weights can be uniform or distance. The uniform weight gives equal weight for each point, while distance gives more weight to closer neighbors. The metric hyperparameter can be several distance metrics. In the scope of this thesis, Euclidean and Manhattan distance metrics are utilized. The Euclidean distance is the closest distance between points, and the Manhattan distance is measured along axes (Pedregosa et al., 2011).

The hyperparameter tuning has also been done for the Naïve Bayes algorithm. The only hyperparameter tuned is the variance smoothing. Variance smoothing indicates that a portion of the highest deviation among all characteristics is added to variances to ensure computation stability (Pedregosa et al., 2011).

The last ML algorithm used in this thesis is the Stochastic Gradient Boosting algorithm. The tuned hyperparameter of this algorithm is the number of estimators, which denotes the number of boosting phases that must be completed. Gradient boosting is somewhat resistant to over-fitting; hence, a big number typically yields better results (Pedregosa et al., 2011).

The five machine learning models already tuned in terms of hyperparameters are established for each intensity measure separately. These models are evaluated in terms of performance metrics: accuracy, precision, recall, and F_1 score. Table 4.1 illustrates each model's performance metrics and the comparisons of these models. Although these performance metrics indicate the general idea of the performance of a model, the confusion matrices are more reliable performance metrics for classification problems.

Appendix C presents the confusion matrices for each tuned model and IM. However, these performance metrics are suitable for the mean predicted result for the data. In the scope of the thesis, each probability of MIDR level for each dataset is predicted, and fragility curves are derived from MIDR level probabilities. Therefore, the training dataset is also used to derive fragility curves with fitting log-normal distribution for each MIDR level. These fragility curves are compared with the fragility curves derived from the ML models.

Table 4.1 illustrates the general idea of how machine learning algorithms are performed under different IMs. Even though these performance metrics contain lots of information on the evaluation of ML algorithms, they cannot be relied upon when deciding which one is the best-fitted algorithm. However, the random forest algorithm has the highest performance metric among the others. Even if the dataset is trying to be generated as a balanced dataset, the count of each MIDR level is not the same or close because MIDRs are calculated after nonlinear time history analysis under different earthquake records. Therefore, accuracy is not a sufficient metric to evaluate the machine learning algorithm. On the other hand, F_1 score, precision, and recall are more reliable metrics for classification problems. Moreover, confusion metrics are used to make detailed investigations through the performance of ML algorithms. The confusion metric for the best-fitted algorithm, which is a random forest algorithm with PGA, is presented in Table 4.2. The rest of the confusion metrics are demonstrated in Appendix C. Additionally, the random forest algorithm also performed well when IM is PGV, PGD, and CAV.

The results in Table 4.2 indicate that predicting a low MIDR level can achieve accuracy with a recall metric ranging from 94% to 74%. Moreover, the random forest algorithm predicts the complete damage value, which refers to the value of MIDR exceeding 1%, with a recall metric of 94%. Nevertheless, MIDR levels between 0.6% and 1% are predicted relatively less correctly than the other MIDR levels.

Table 4.1. Evaluation metrics for the test data of machine learning algorithms are given for the specified IM.

Classification Algorithm	IM	Precision	Recall	F ₁ Score	Accuracy
Decision Tree (DT)	PGA	0.69	0.69	0.69	0.77
	PGV	0.69	0.69	0.69	0.77
	PGD	0.68	0.68	0.68	0.77
	Sa(T ₁)	0.39	0.38	0.38	0.52
	Sd(T ₁)	0.38	0.37	0.37	0.52
	CAV	0.69	0.69	0.69	0.77
Random Forest (RF)	PGA	0.79	0.78	0.78	0.85
	PGV	0.78	0.78	0.78	0.84
	PGD	0.78	0.78	0.78	0.84
	Sa(T ₁)	0.43	0.41	0.42	0.55
	Sd(T ₁)	0.42	0.40	0.41	0.55
	CAV	0.78	0.78	0.78	0.84
K-Nearest Neighbor (KNN)	PGA	0.29	0.28	0.27	0.45
	PGV	0.27	0.26	0.25	0.43
	PGD	0.24	0.23	0.22	0.39
	Sa(T ₁)	0.34	0.33	0.33	0.49
	Sd(T ₁)	0.34	0.33	0.33	0.49
	CAV	0.32	0.30	0.30	0.47
Naïve Bayes (NB)	PGA	0.27	0.25	0.23	0.42
	PGV	0.25	0.24	0.21	0.41
	PGD	0.21	0.22	0.19	0.39
	Sa(T ₁)	0.26	0.26	0.24	0.43
	Sd(T ₁)	0.30	0.31	0.29	0.48
	CAV	0.24	0.25	0.23	0.43
Stochastic Gradient Boosting (SGB)	PGA	0.47	0.41	0.41	0.59
	PGV	0.48	0.40	0.40	0.58
	PGD	0.44	0.36	0.36	0.55
	Sa(T ₁)	0.35	0.33	0.32	0.52
	Sd(T ₁)	0.35	0.34	0.33	0.53
	CAV	0.48	0.41	0.41	0.58

Table 4.2. Confusion matrix of RF algorithm with PGA.

RF – PGA		Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
Actual Class		0.0% - 0.1%	1391	94	1	0	0	0	0	0	0	0	1486	0.94
		0.1% - 0.2%	96	2784	123	1	0	3	0	0	0	1	3008	0.93
		0.2% - 0.3%	1	133	2023	142	3	0	2	2	0	0	2308	0.88
		0.3% - 0.4%	0	5	144	1595	163	2	5	0	2	0	1920	0.83
		0.4% - 0.5%	0	0	3	156	1162	114	3	3	1	1	1456	0.80
		0.5% - 0.6%	0	1	5	4	147	1002	155	15	4	1	1358	0.74
		0.6% - 0.7%	0	0	3	2	5	164	809	142	26	3	1194	0.68
		0.7% - 0.8%	0	0	2	0	0	15	117	609	101	13	903	0.67
		0.8% - 0.9%	0	0	0	2	0	1	10	112	510	102	809	0.63
		0.9% - 1.0%	0	0	0	0	1	2	3	10	89	395	665	0.59
>1.0%		0	0	2	1	10	19	28	42	63	144	4955	5264	0.94
Total		1488	3017	2306	1903	1491	1322	1132	935	796	661	5320	0.85	
Precision		0.93	0.92	0.88	0.84	0.78	0.76	0.71	0.65	0.64	0.60	0.93		

When the other algorithms and IM metrics are considered, the model that best predicts the MIDR exceeding 1% is the Stochastic Gradient Boosting algorithm and PGA pair with a recall metric of 96%. However, this algorithm gives less performance when predicting the other MIDR levels. The decision tree algorithm is the second-best algorithm after the random forest algorithm. This algorithm is also good at distinguishing each MIDR level from the others. However, this algorithm cannot efficiently predict mid-range MIDR levels.

Naïve Bayes and KNN algorithms are the worst algorithms for this dataset; however, the performance of these algorithms can be improved with over-sampling methods, under-sampling methods, or generating the dataset by using the features' distribution (Kaur et al., 2020; Ulku et al., 2022).

Another noteworthy observation from the evaluation metrics is that each algorithm exhibited improved performance when PGA, PGV, PGD, or CAV were utilized as intensity measures (IM) in the dataset. This tendency may be attributed to the interdependence between $S_a(T_1)$, $S_d(T_1)$, and the selected ground motion. Strong ground motions are selected by employing the conditional mean spectrum, which is conditioned on the first fundamental vibration period. Each data point exhibits a dependency on $S_a(T_1)$ or $S_d(T_1)$ when utilized as IM in the dataset. Consequently, this dependency may lead these algorithms to produce less reliable results.

In this thesis, the probability of each MIDR level is predicted for each data point. These probabilities are used to derive fragility curves. Figures 4.1, 4.2, 4.3, 4.4, and 4.5 illustrate an example of the probability mass function for a data point for each algorithm when the dataset is generated with PGA. This process is applied to each data point in the dataset. Then, the total probability theorem is used to find the probability of exceedance given MIDR at a specific IM bin.

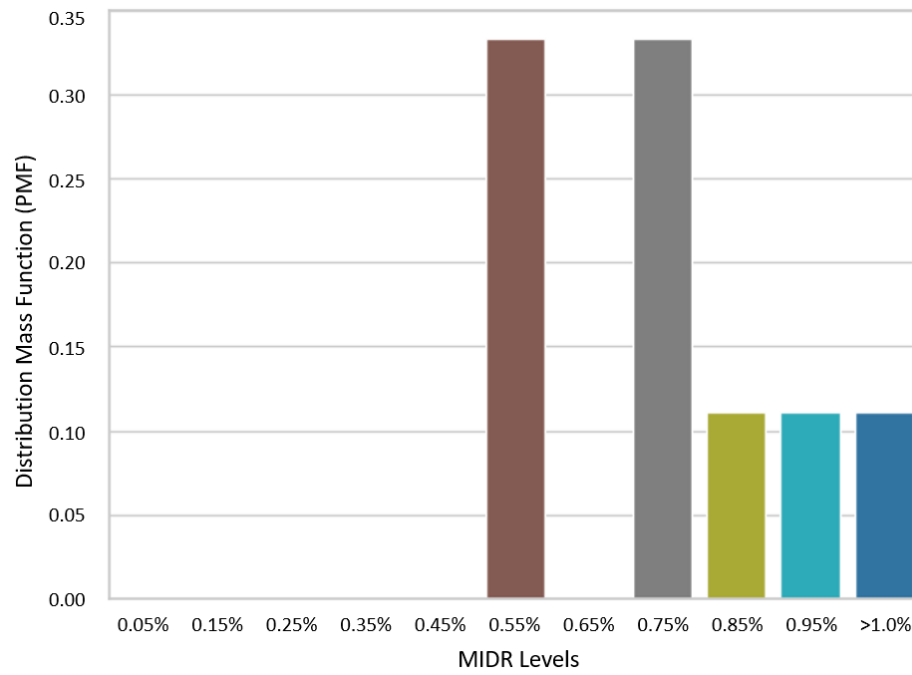


Figure 4.1. The probability mass function of MIDR levels for a random data point, dataset with PGA, obtained from the Decision Tree algorithm.

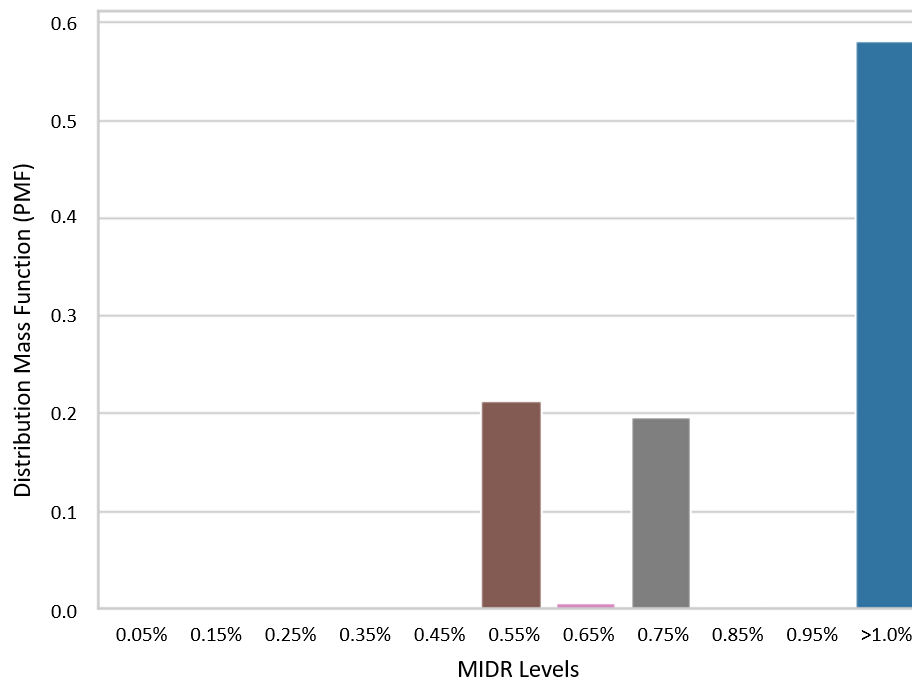


Figure 4.2. The probability mass function of MIDR levels for a random data point, dataset with PGA, obtained from the Random Forest algorithm.

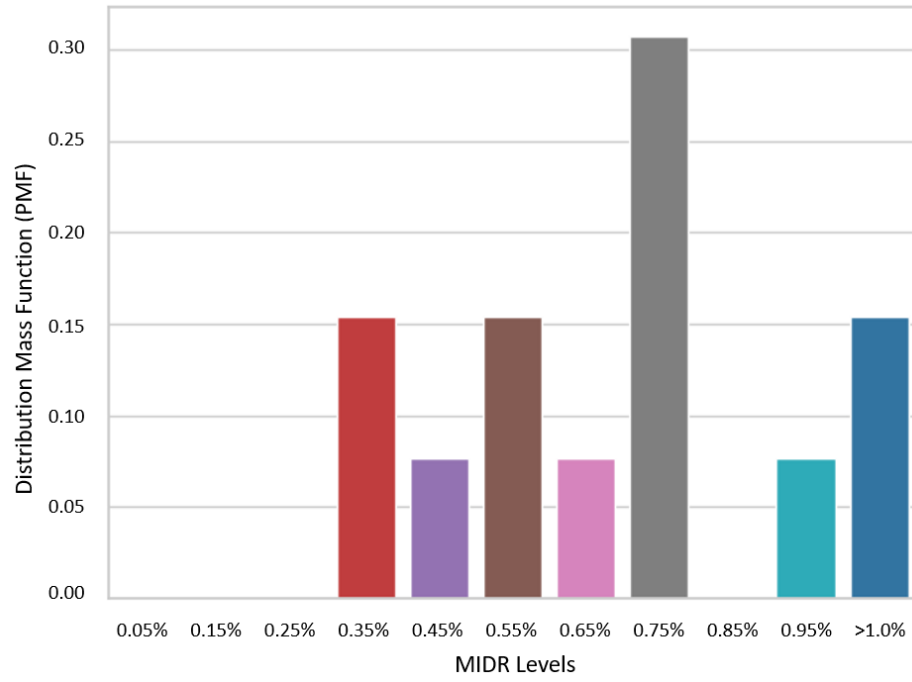


Figure 4.3. The probability mass function of MIDR levels for a random data point, dataset with PGA obtained from the KNN algorithm.

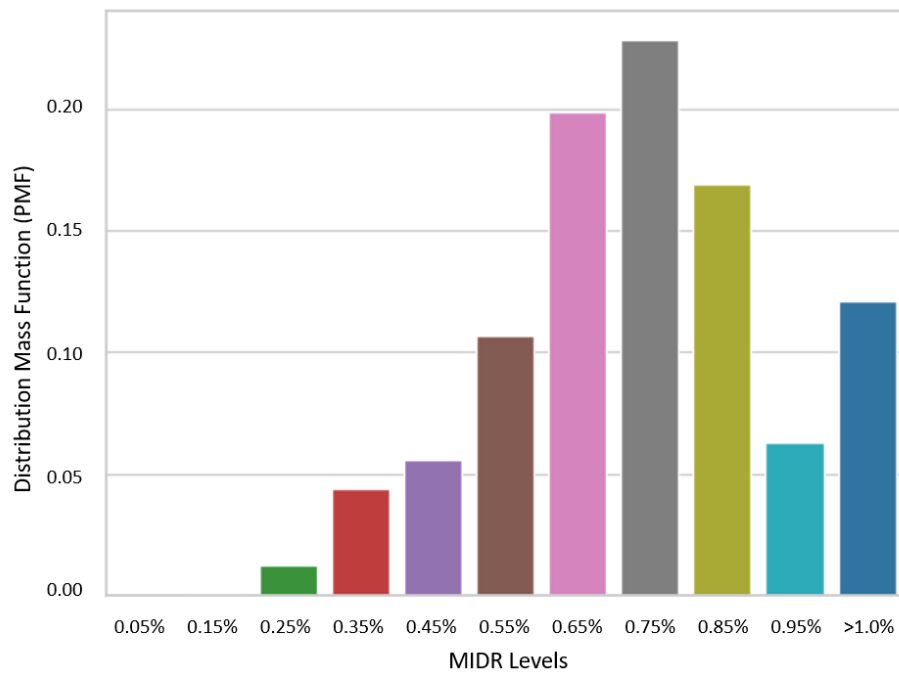


Figure 4.4. The probability mass function of MIDR levels for a random data point, dataset with PGA obtained from the Naïve Bayes algorithm.

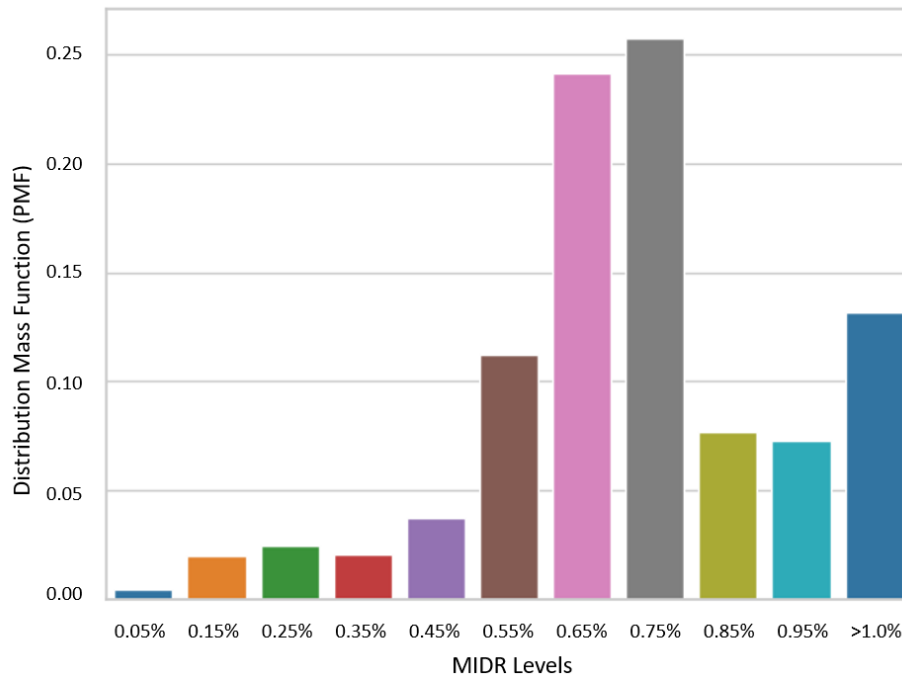


Figure 4.5. The probability mass function of MIDR levels for a random data point, dataset with PGA obtained from the Stochastic Gradient Boosting algorithm.

4.1. Derivation of Fragility Curves via ML Algorithms

The probability of each MIDR level for each data point is predicted via ML algorithms. These data points are classified through the IM bin. Then, Equation 3.2 is applied to each MIDR level and range of IM values. Figures between 4.6 and 4.35 illustrate fragility curves derived from the Decision Tree, Random Forest, KNN, Naïve Bayes, and Stochastic Gradient Boosting algorithms with respect to PGA, PGV, PGD, $S_a(T_1)$, $S_d(T_1)$, and CAV.

Since Decision Tree, Random Forest, and Stochastic Gradient Boosting algorithms are types of tree classification, the fragility curves appear as a step function. This step function is a consequence of the if-else statements while predicting the real class. This step function is an actual predicted cumulative distribution function for each MIDR level. Therefore, these fragility curves tend to show the actual fragility curves in terms of the utilized dataset.

On the contrary, the KNN algorithm is smoother than the Decision Tree, Random Forest, and Stochastic Gradient Boosting algorithms. KNN is non-parametric supervised learning and classifies the output according to the voting system. Thus, the distinction between classes can be expressed as a function. This function makes the fragility smoother. Furthermore, the Naïve Bayes algorithm also has smooth fragility curves due to predicting the maximum posterior probability. In this case, each class is assumed to be distributed as a Gaussian distribution. One of the reasons for the standardization of the data for the Naïve Bayes algorithm is to utilize the continuous normal (Gaussian) distribution. Thus, these MIDR levels, with respect to standardized IMs, which are converted with a power transformation, are normally distributed. However, when IM is converted back to the actual distribution, the distribution of exceedance probability of specific MIDR level also changes to this actual IM distribution. The reason for changing the distribution of exceedance probability of specific MIDR level is that the fragility curves depend on the IM distribution.

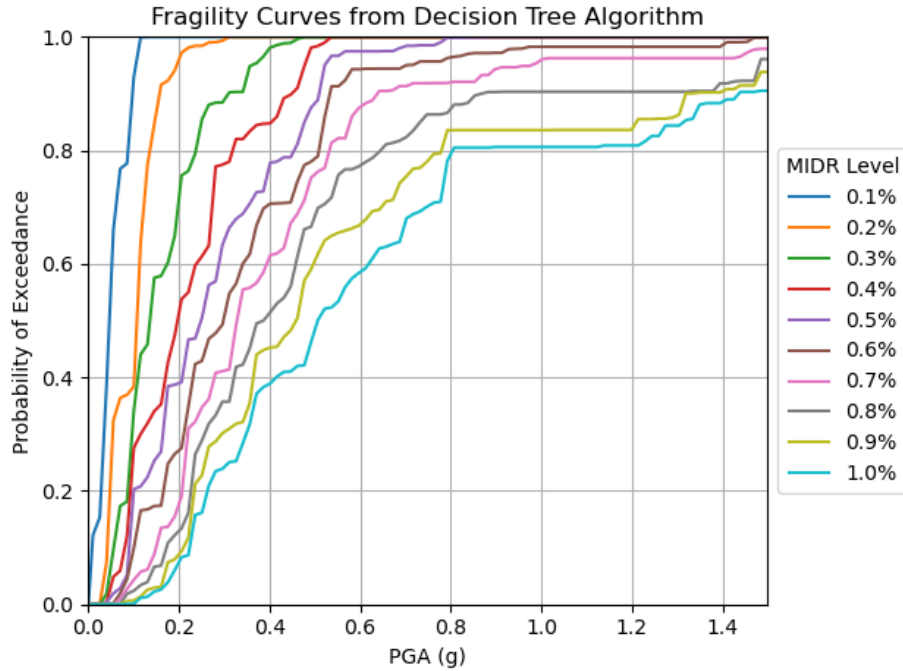


Figure 4.6. Fragility curves derived from the Decision Tree algorithm for PGA.

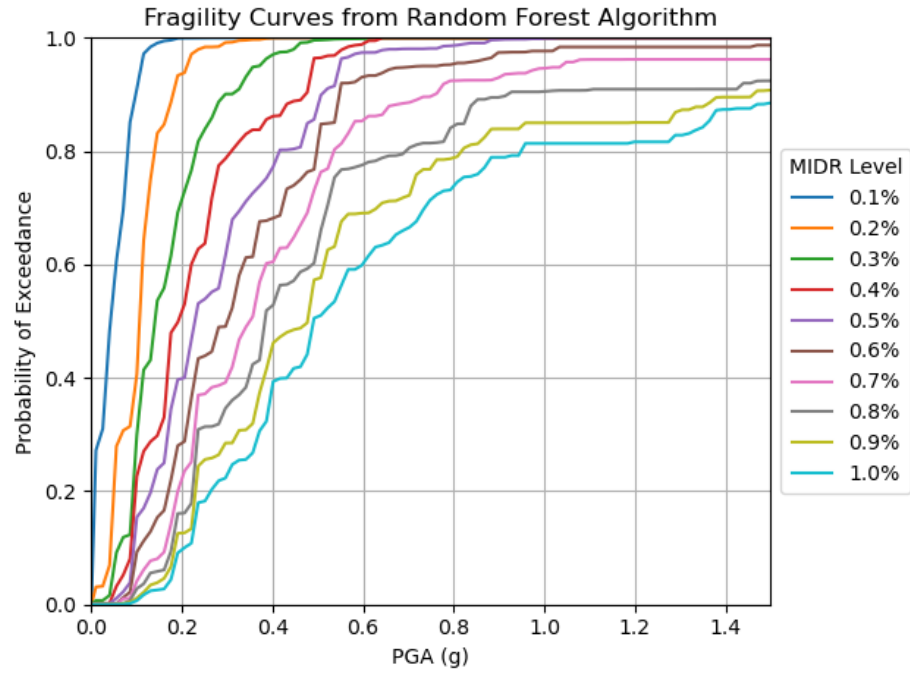


Figure 4.7. Fragility curves derived from the Random Forest algorithm for PGA.

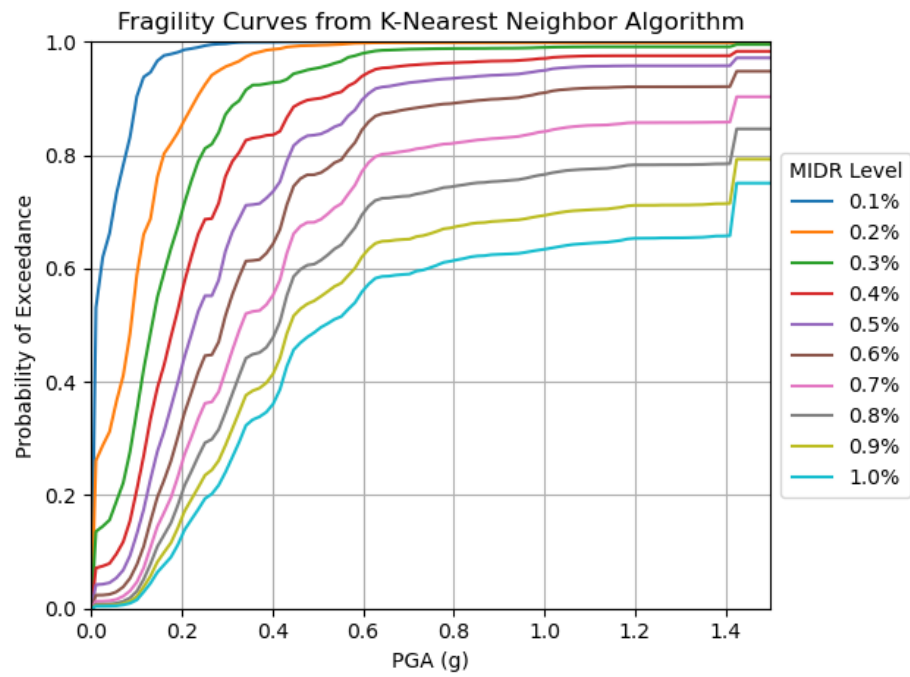


Figure 4.8. Fragility curves derived from the KNN algorithm for PGA.

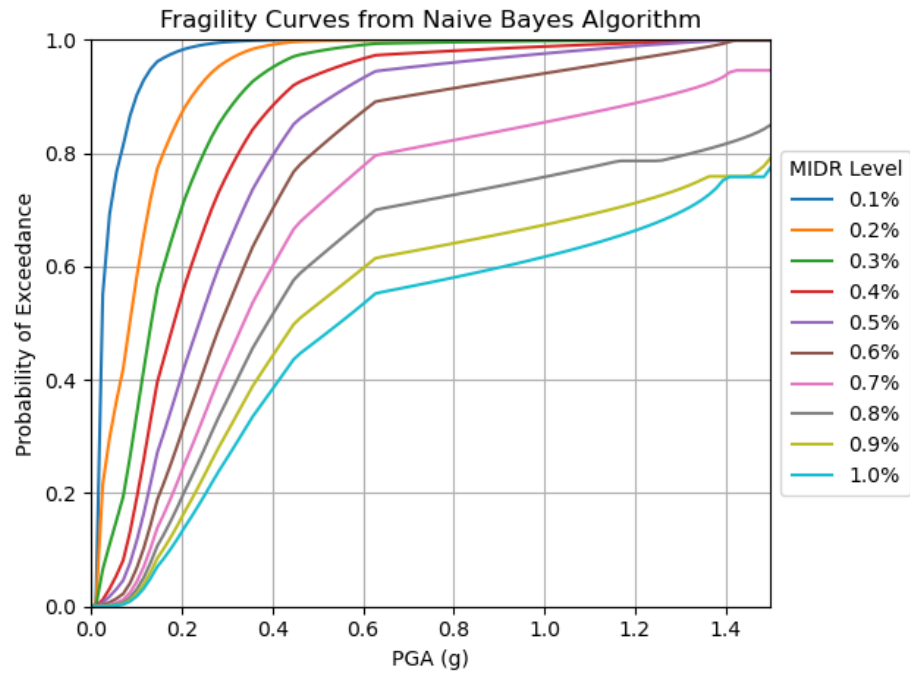


Figure 4.9. Fragility curves derived from the Naïve Bayes algorithm for PGA.

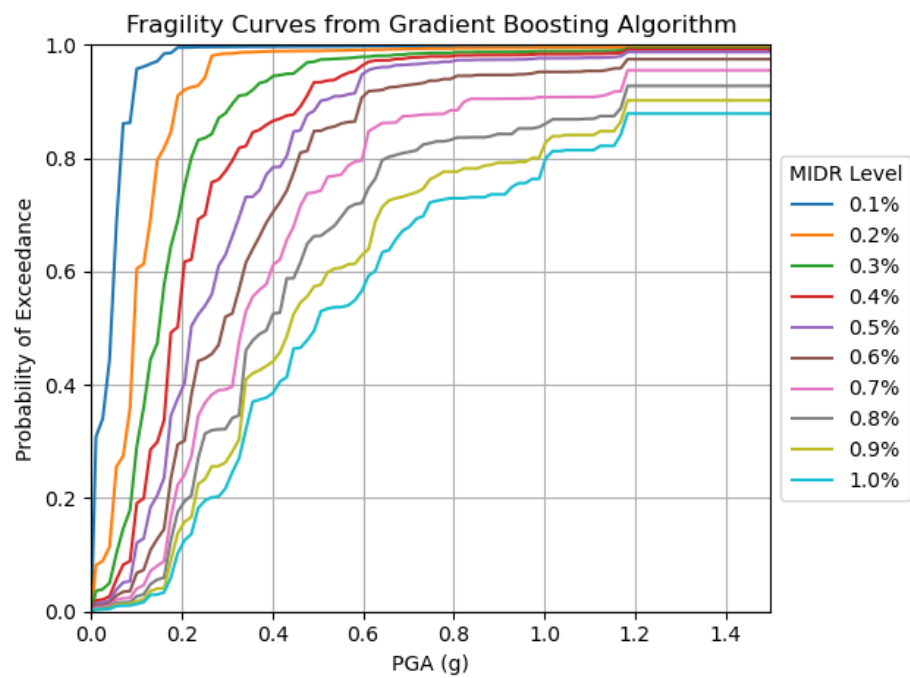


Figure 4.10. Fragility curves derived from the Stochastic Gradient Boosting algorithm for PGA.

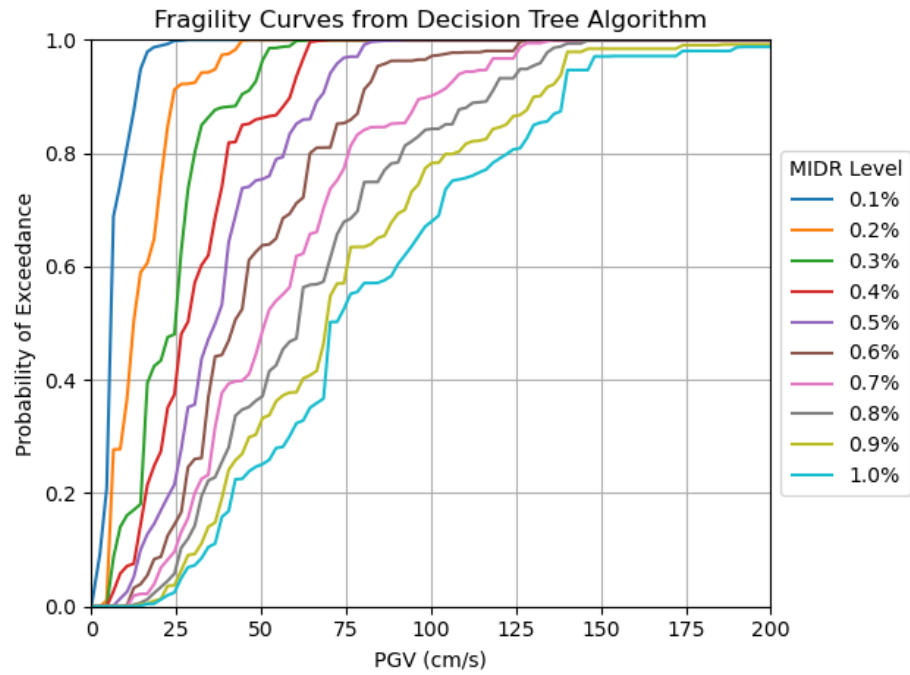


Figure 4.11. Fragility curves derived from the Decision Tree algorithm for PGV.

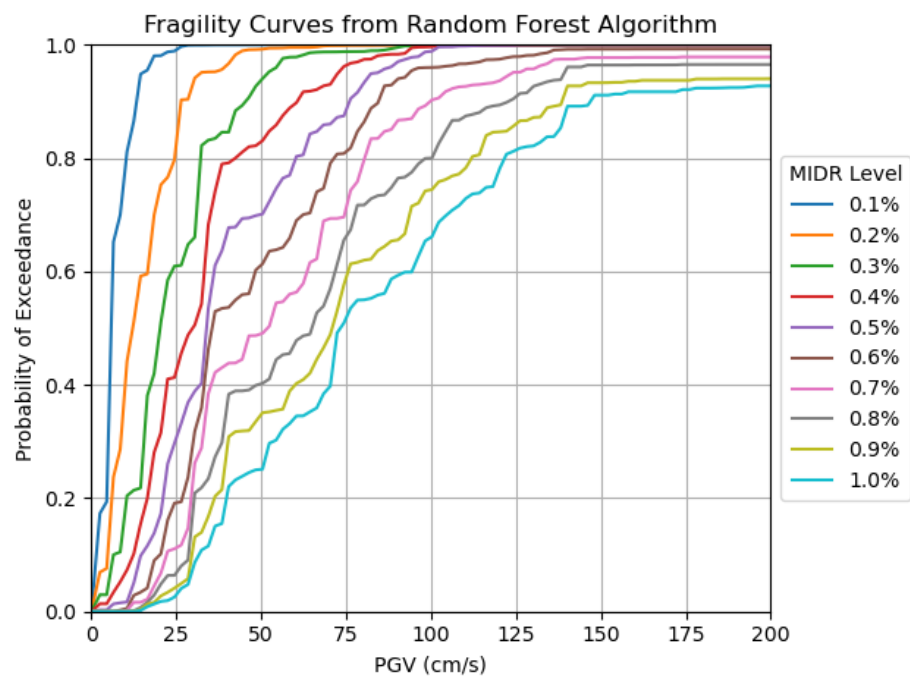


Figure 4.12. Fragility curves derived from the Random Forest algorithm for PGV.

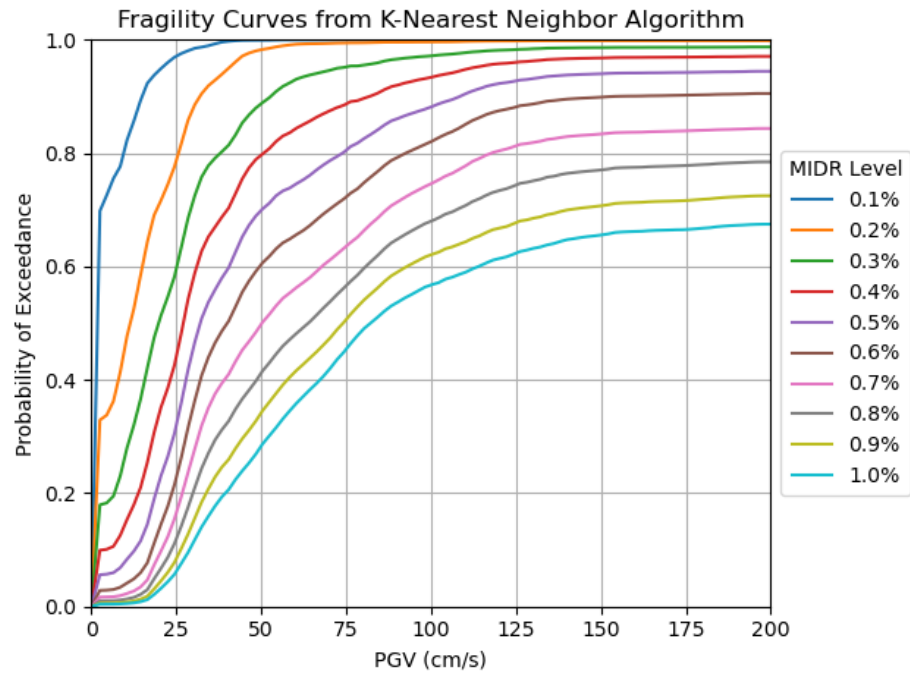


Figure 4.13. Fragility curves derived from the KNN algorithm for PGV.

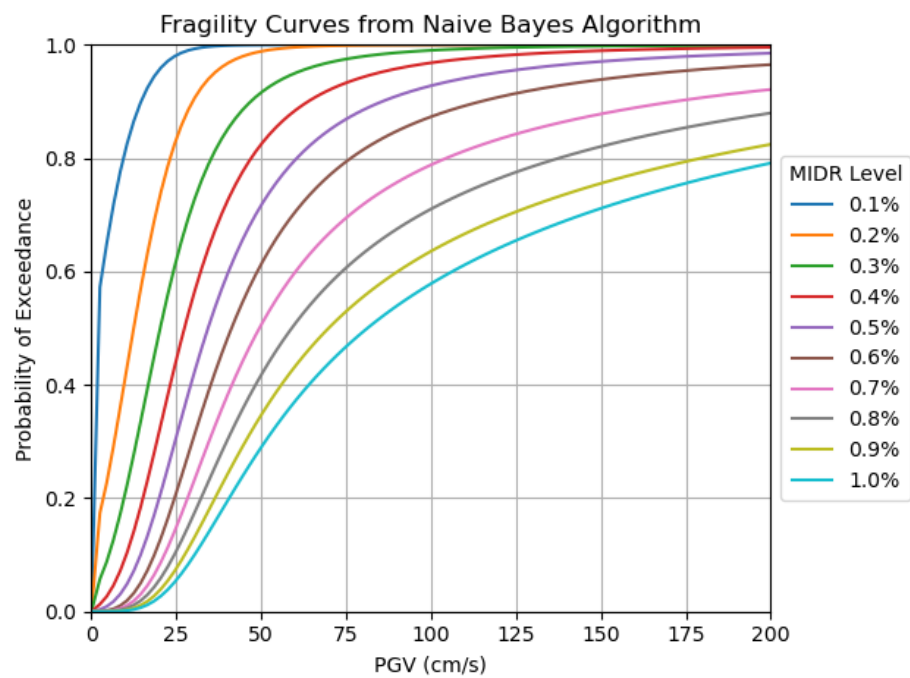


Figure 4.14. Fragility curves derived from the Naïve Bayes algorithm for PGV.

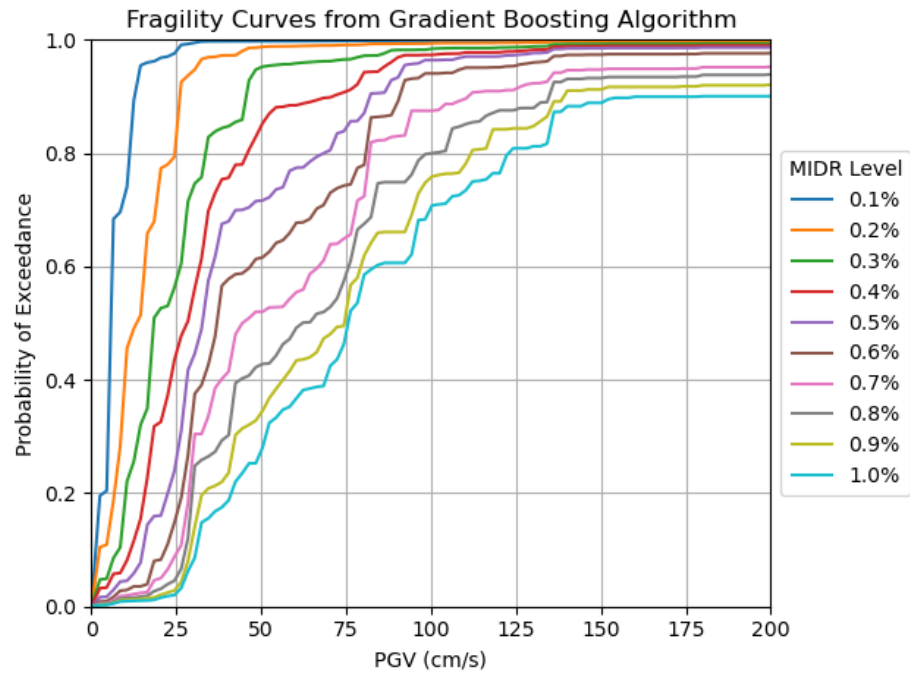


Figure 4.15. Fragility curves derived from the Stochastic Gradient Boosting algorithm for PGV.

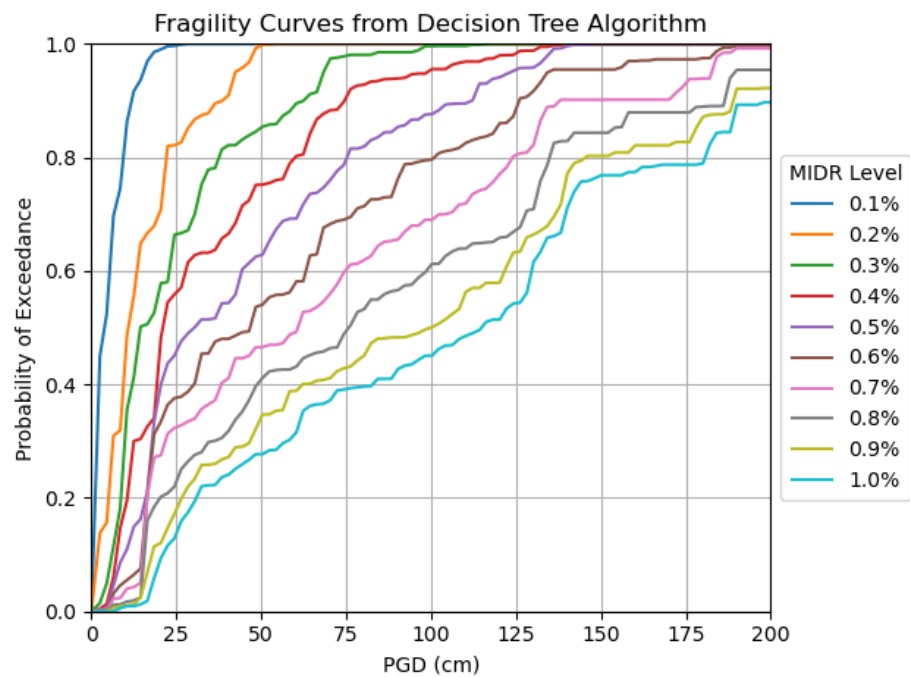


Figure 4.16. Fragility curves derived from the Decision Tree algorithm for PGD.

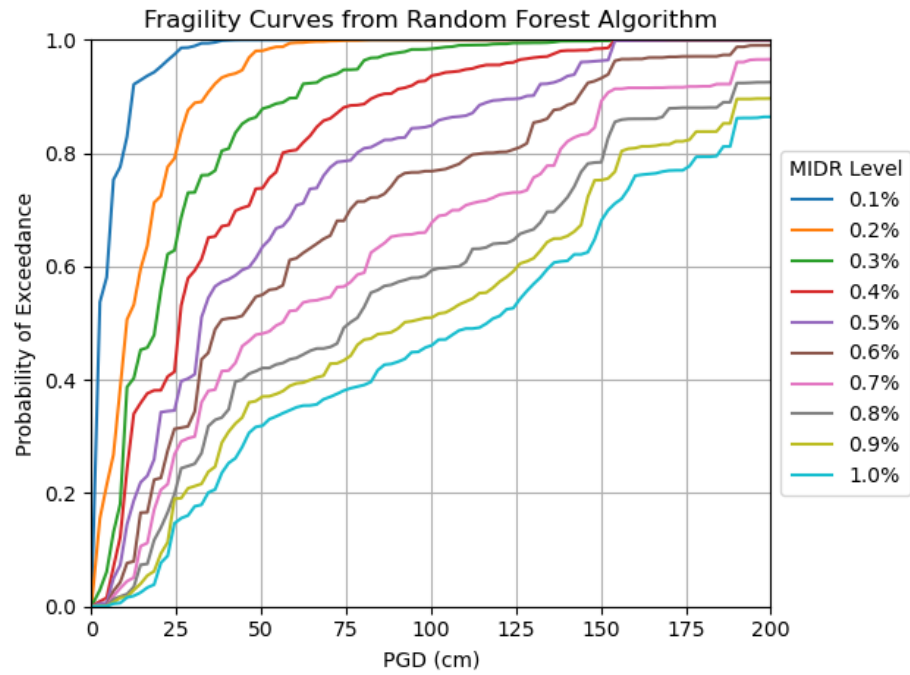


Figure 4.17. Fragility curves derived from the Random Forest algorithm for PGD.

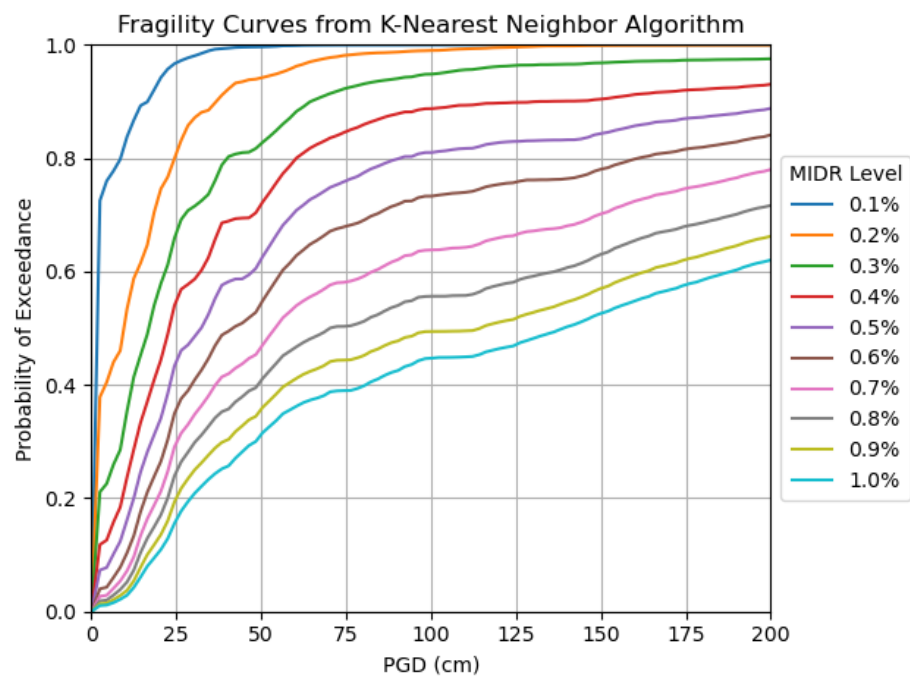


Figure 4.18. Fragility curves derived from the KNN algorithm for PGD.

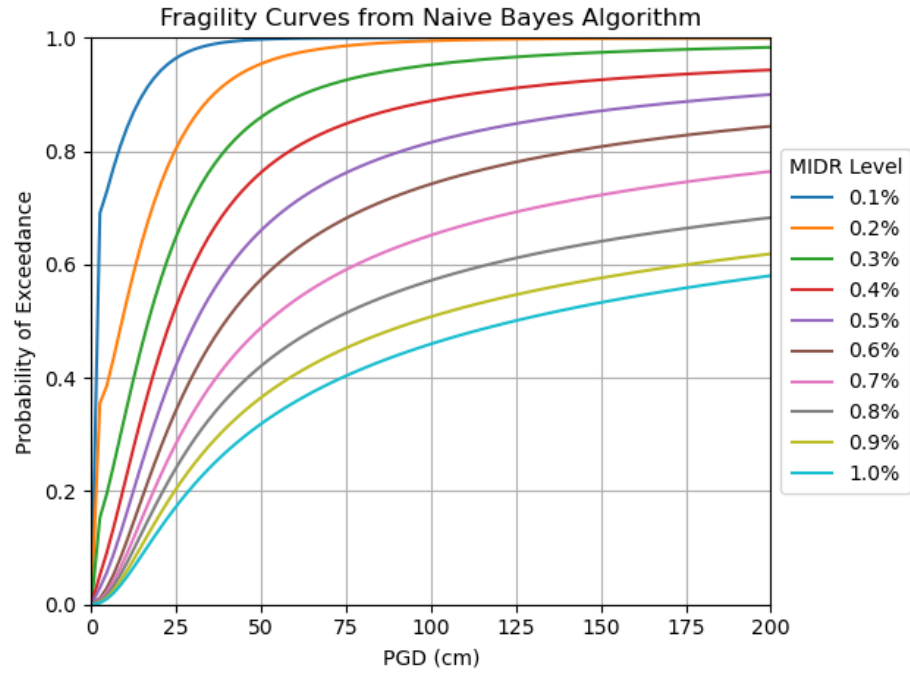


Figure 4.19. Fragility curves derived from the Naïve Bayes algorithm for PGD.

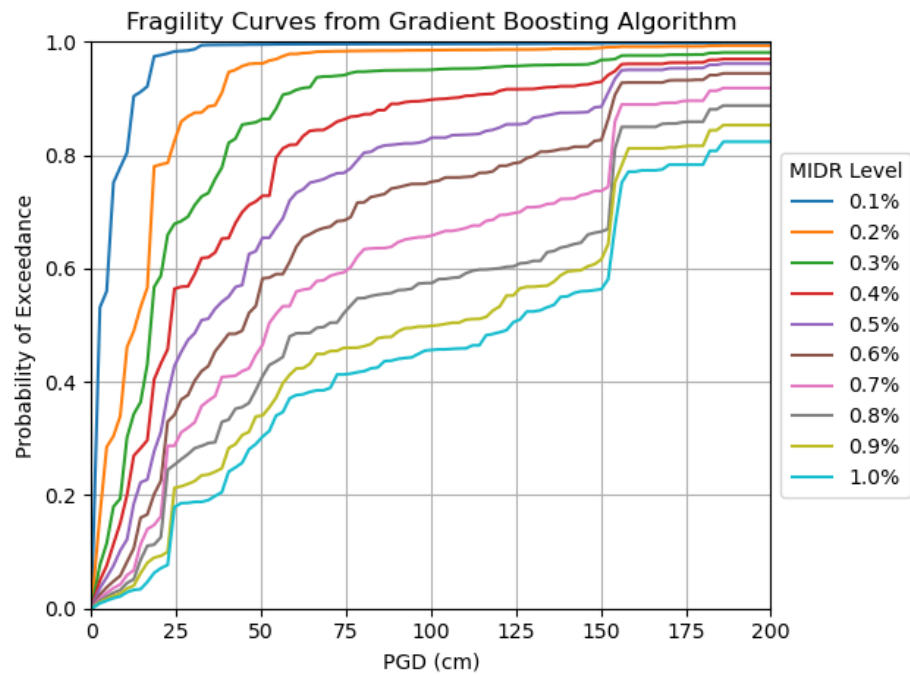


Figure 4.20. Fragility curves derived from the Stochastic Gradient Boosting algorithm for PGD.

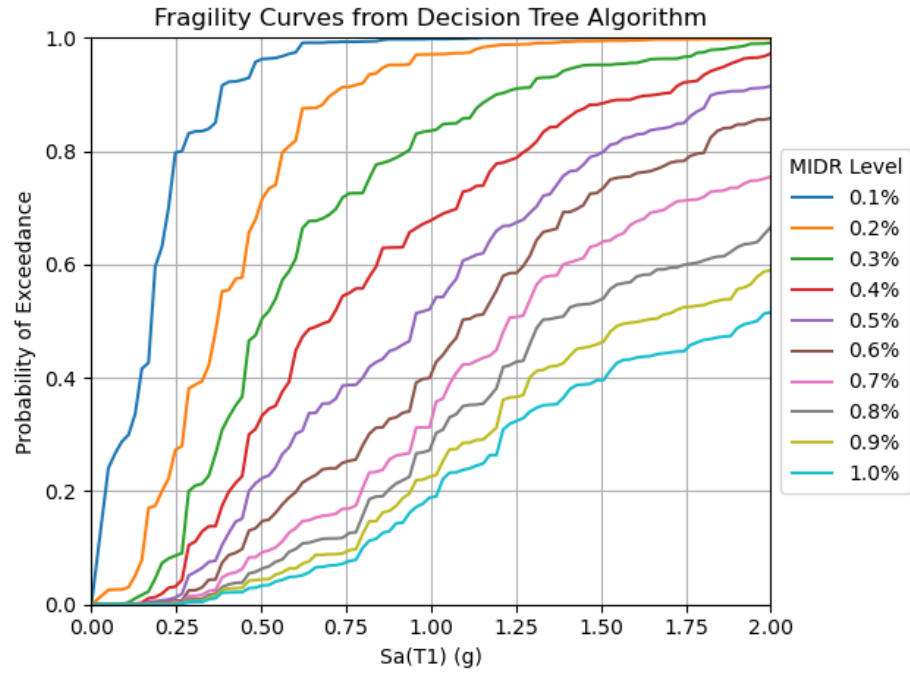


Figure 4.21. Fragility curves derived from the Decision Tree algorithm for $S_a(T_1)$.

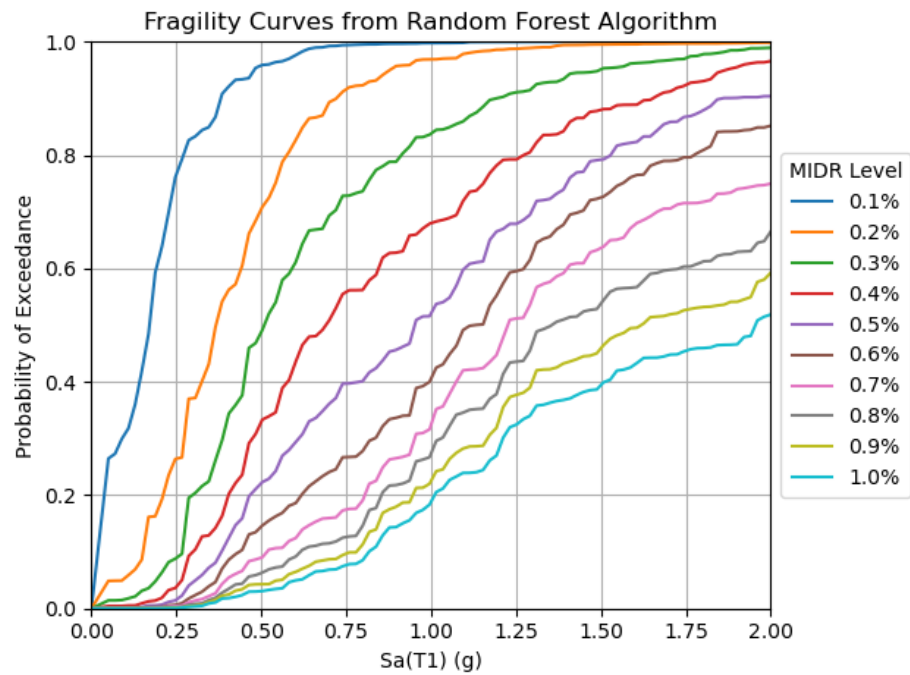


Figure 4.22. Fragility curves derived from the Random Forest algorithm for $S_a(T_1)$.

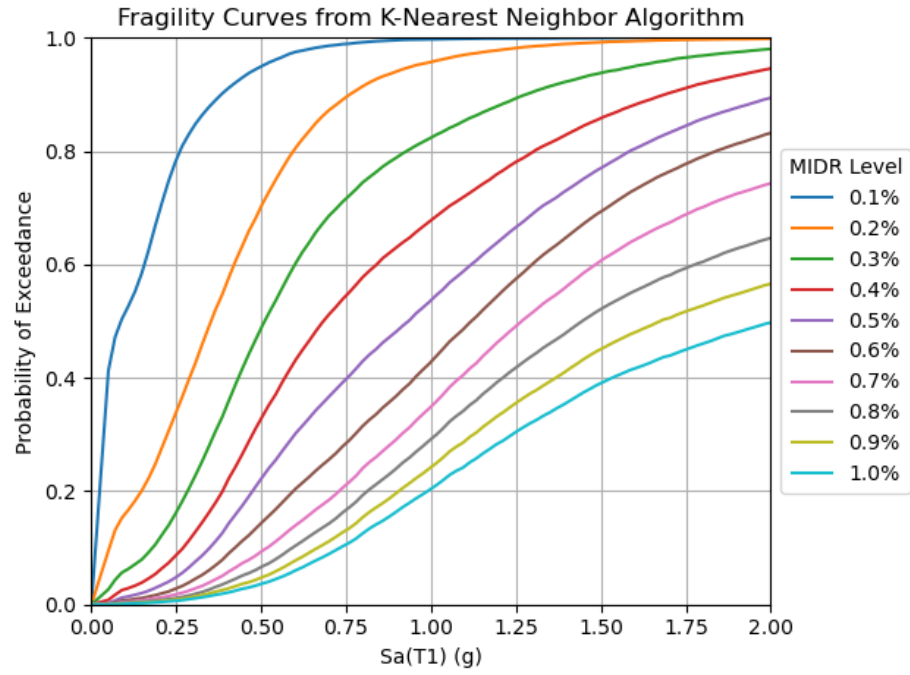


Figure 4.23. Fragility curves derived from the KNN algorithm for $S_a(T_1)$.

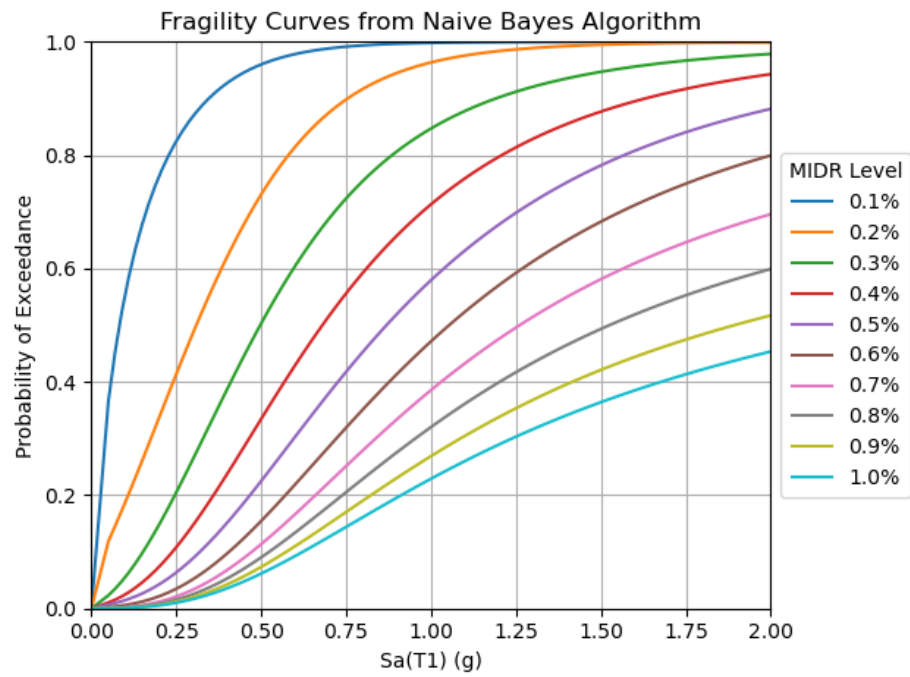


Figure 4.24. Fragility curves derived from the Naïve Bayes algorithm for $S_a(T_1)$.

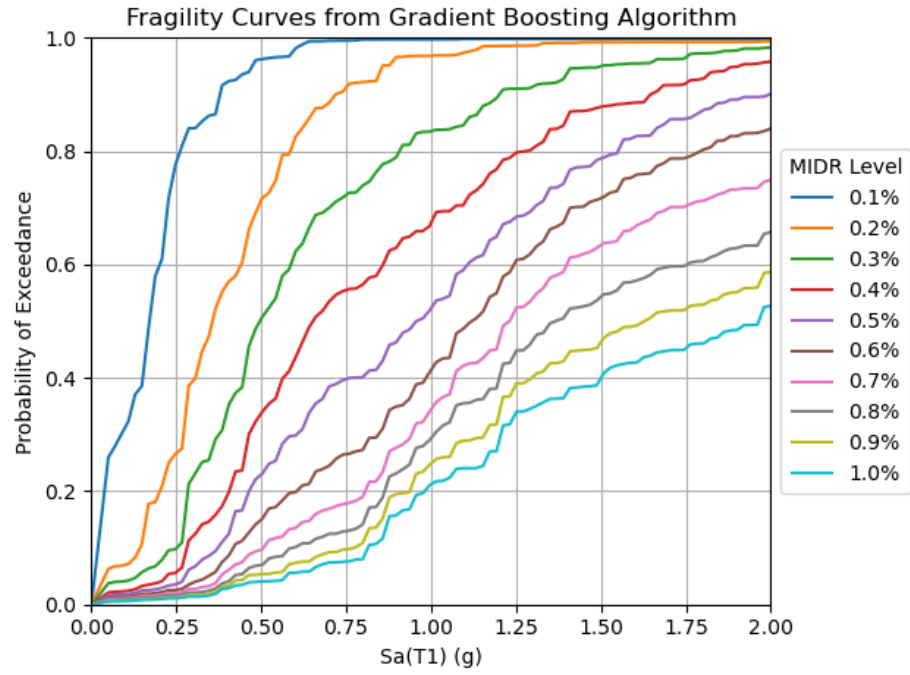


Figure 4.25. Fragility curves derived from the Stochastic Gradient Boosting algorithm for $S_a(T_1)$.

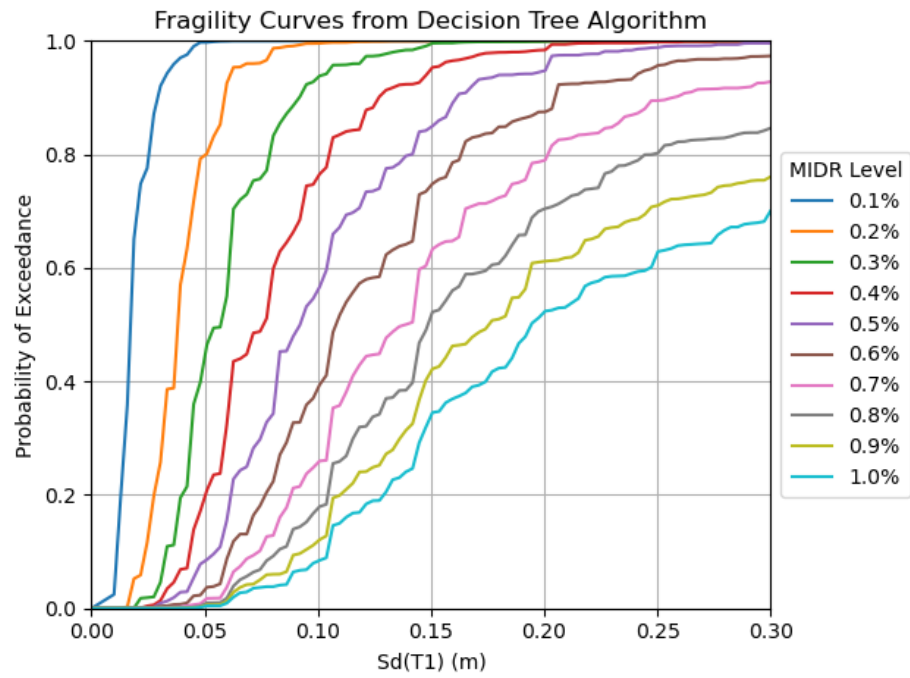


Figure 4.26. Fragility curves derived from the Decision Tree algorithm for $S_d(T_1)$.

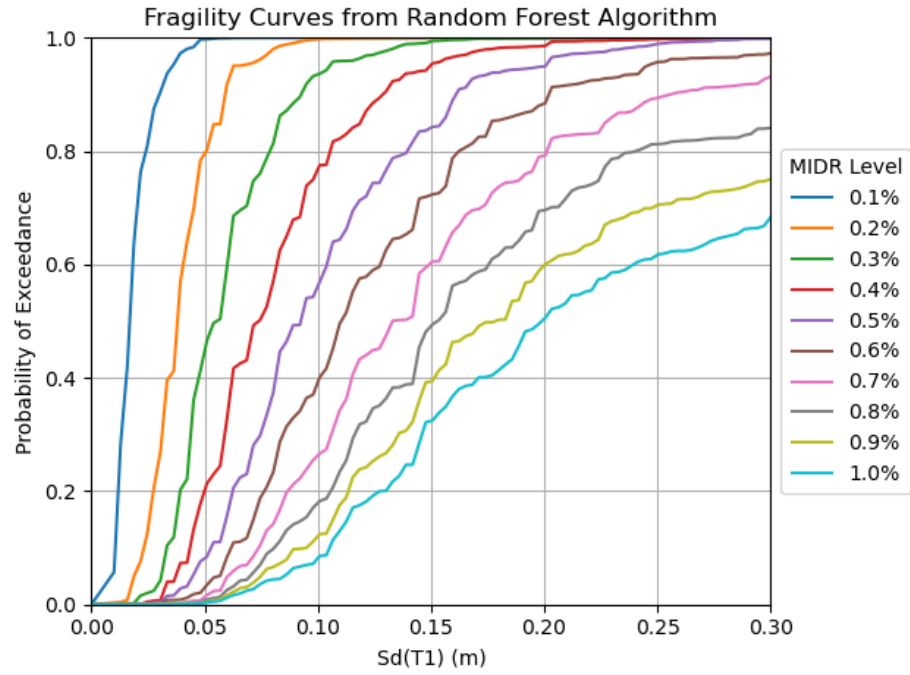


Figure 4.27. Fragility curves derived from the Random Forest algorithm for $S_d(T_1)$.

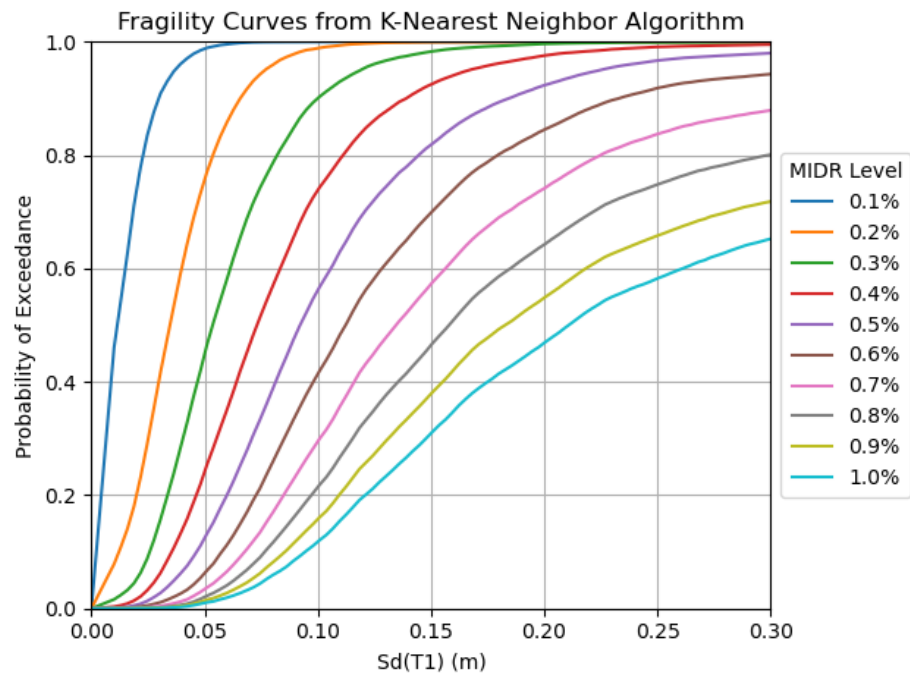


Figure 4.28. Fragility curves derived from the KNN algorithm for $S_d(T_1)$.

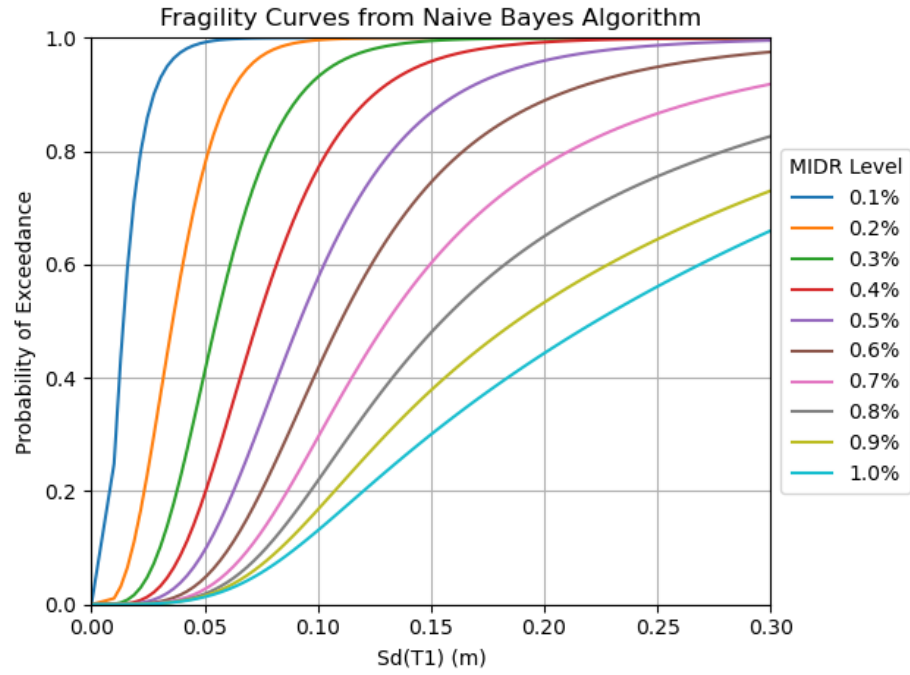


Figure 4.29. Fragility curves derived from the Naïve Bayes algorithm for $S_d(T_1)$.

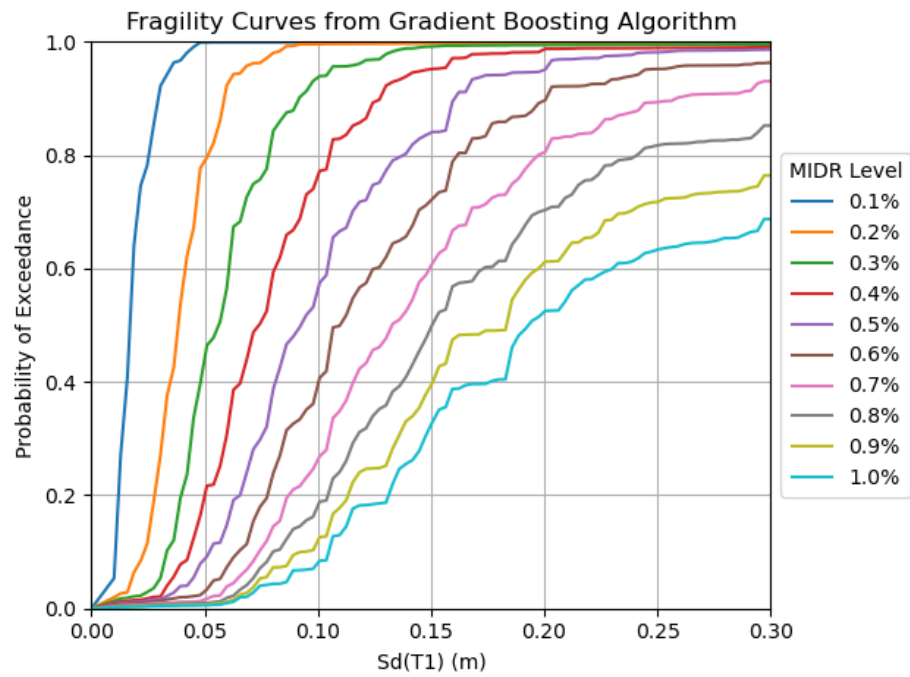


Figure 4.30. Fragility curves derived from the Stochastic Gradient Boosting algorithm for $S_d(T_1)$.

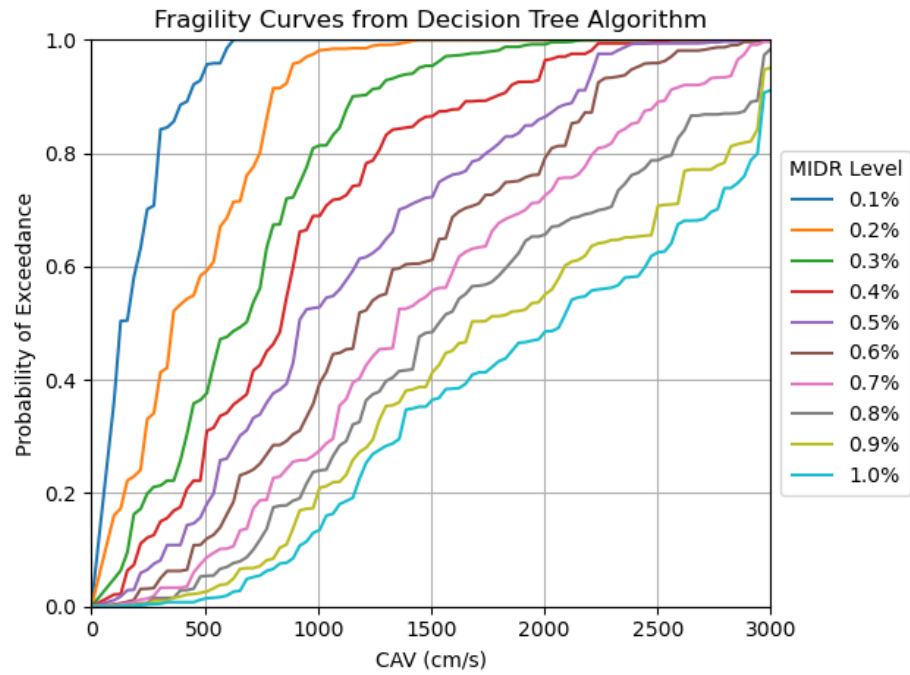


Figure 4.31. Fragility curves derived from the Decision Tree algorithm for CAV.

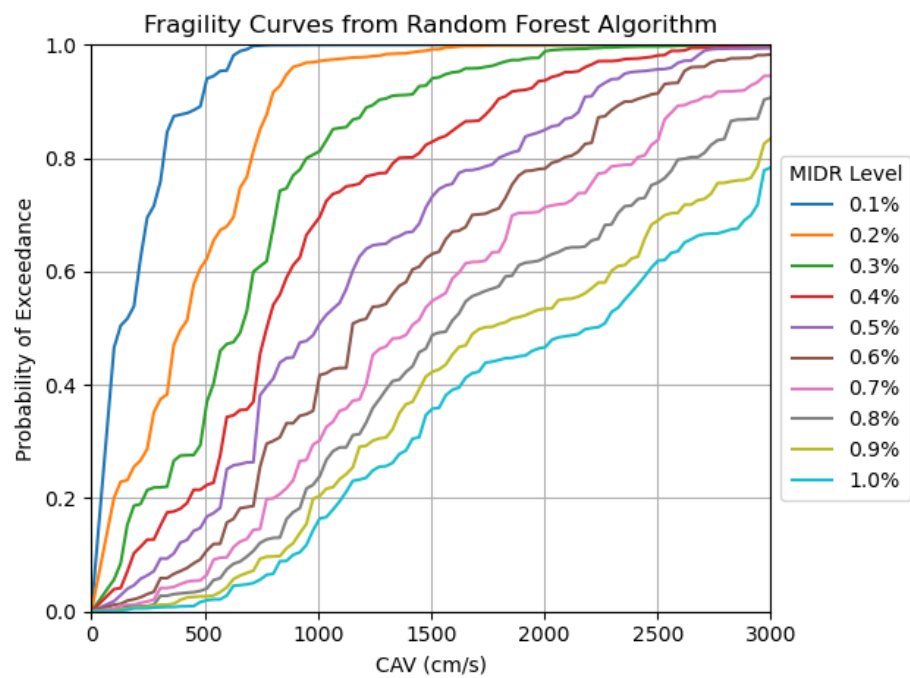


Figure 4.32. Fragility curves derived from the Random Forest algorithm for CAV.

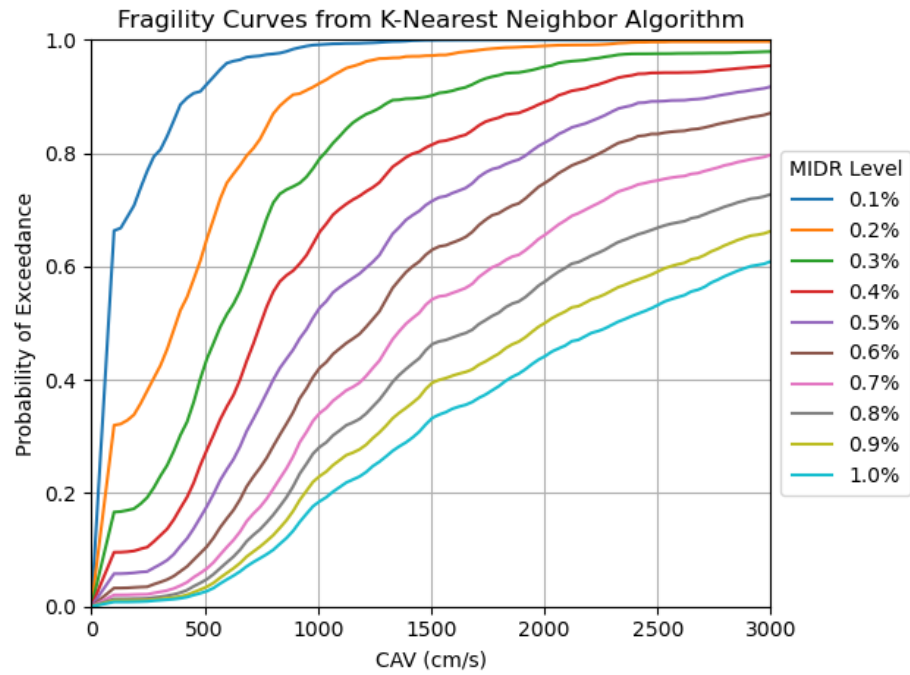


Figure 4.33. Fragility curves derived from the KNN algorithm for CAV.

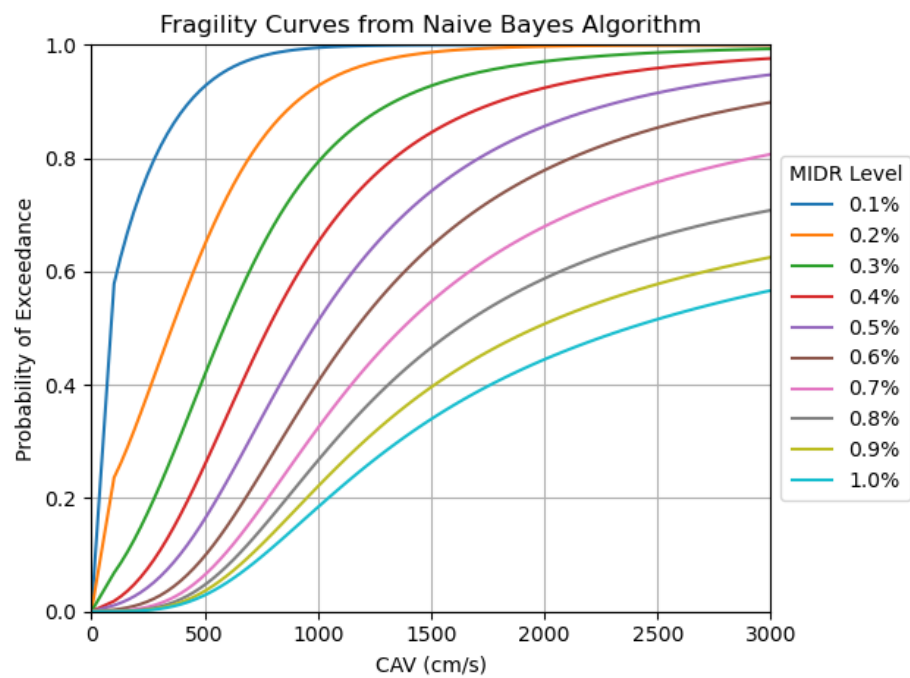


Figure 4.34. Fragility curves derived from the Naïve Bayes algorithm for CAV.

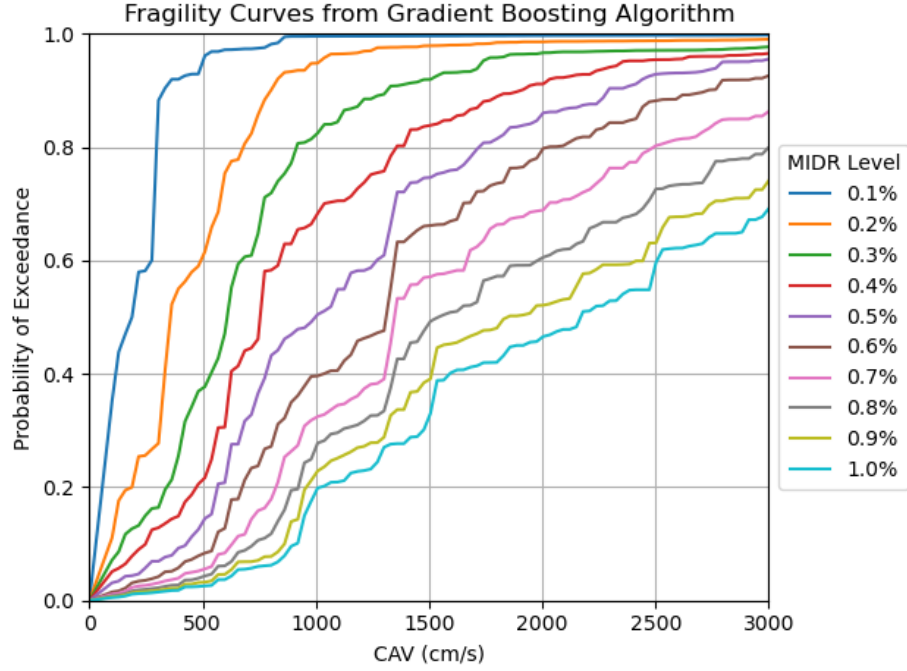


Figure 4.35. Fragility curves derived from the Stochastic Gradient Boosting algorithm for CAV.

4.2. Derivation of Fragility Curves Through Log-normal Distribution

Shinozuka et al. (2000) presented an approach to generating fragility curves. This approach utilizes the maximum likelihood estimation (MLE) method to predict the parameters of log-normal distribution. The initial hypothesis that fragility curves are log-normally distributed originated from research done for risk assessment for nuclear power plants (Kennedy et al., 1980; Kennedy and Ravindra, 1984). Many researchers prefer this method due to its simplicity, which only requires two parameters (mean and standard deviation). However, other approaches can be used to derive fragility curves, such as machine learning, fitting logistic regression, or Bayesian probability.

Baker (2015) elucidates the process of the MLE method while deriving fragility curves. Equation 4.1 shows the log-normal cumulative distribution function for a spe-

cific MIDR level and IM.

$$P(MIDR \geq midr_i | IM = x) = \Phi\left(\frac{\ln(x/\theta_i)}{\beta_i}\right) \quad (4.1)$$

where $P(MIDR \geq midr_i | IM = x)$ is the exceedance probability of a specific MIDR at ground motion with $IM = x$. $\Phi()$ is the standard normal cumulative distribution function, θ_i is the median of the fragility curves specified with a MIDR level, and β is the standard deviation of the $\ln(IM)$ for that specific MIDR level.

The MLE method is used to predict the median and standard deviation of the best-fitted log-normal distribution for a specific MIDR level. However, before employing this method, the number of buildings that exceed the specific MIDR level and the total number of buildings for a specific IM level is determined. The probability of observing the specific MIDR level at a particular IM level from the numbers given above is then calculated using the binomial distribution. Equation 4.2 expresses the binomial distribution function.

$$P(k_j, n_j, p_j) = \binom{n_j}{k_j} p_j^{k_j} (1 - p_j)^{n_j - k_j} \quad (4.2)$$

where j is the particular level of IM, k_j is the number of buildings exceeding the specific MIDR level at that IM level, n_j is the number of total buildings at that IM level, and p_j is the probability of exceedance probability of MIDR level at that level of IM. This probability is calculated by utilizing log-normal distribution, and likelihood is maximized by setting the log-normal distribution parameters using the maximum likelihood method.

When using multiple intensity measure levels, the likelihood function, which is the product of the binomial probabilities, may be represented as

$$Likelihood = \prod_j^m \binom{n_j}{k_j} p_j^{k_j} (1 - p_j)^{n_j - k_j} \quad (4.3)$$

If we replace p_j by the cumulative distribution function (CDF) of the log-normal distribution, Equation 4.3 becomes Equation 4.5. If we take the log of the random variables of the log-normal distribution, the log-normal distribution becomes the normal distribution. Thus, the CDF of the normal distribution can be used, as shown in

Equation 4.5.

$$Likelihood = \prod_j^m \binom{n_j}{k_j} \Phi\left(\frac{\ln(x/\theta)}{\beta}\right)^{k_j} \left(1 - \Phi\left(\frac{\ln(x/\theta)}{\beta}\right)\right)^{n_j - k_j} \quad (4.4)$$

The last step is to maximize the likelihood by setting the parameters of θ and β . So, this function can be expressed as

$$\{\hat{\theta}, \hat{\beta}\} = \arg \max_{\theta, \beta} \sum_{j=1}^m \left\{ \ln \binom{n_j}{k_j} + k_j \ln \Phi\left(\frac{\ln(x/\theta)}{\beta}\right) + (n_j - k_j) \ln \left(1 - \Phi\left(\frac{\ln(x/\theta)}{\beta}\right)\right) \right\} \quad (4.5)$$

The fragility curves by utilizing the maximum likelihood method are derived as a function of different IMs and illustrated in figures between 4.36 and 4.41. These fragility curves are compared with the fragility curves derived via ML algorithms in Section 4.3.

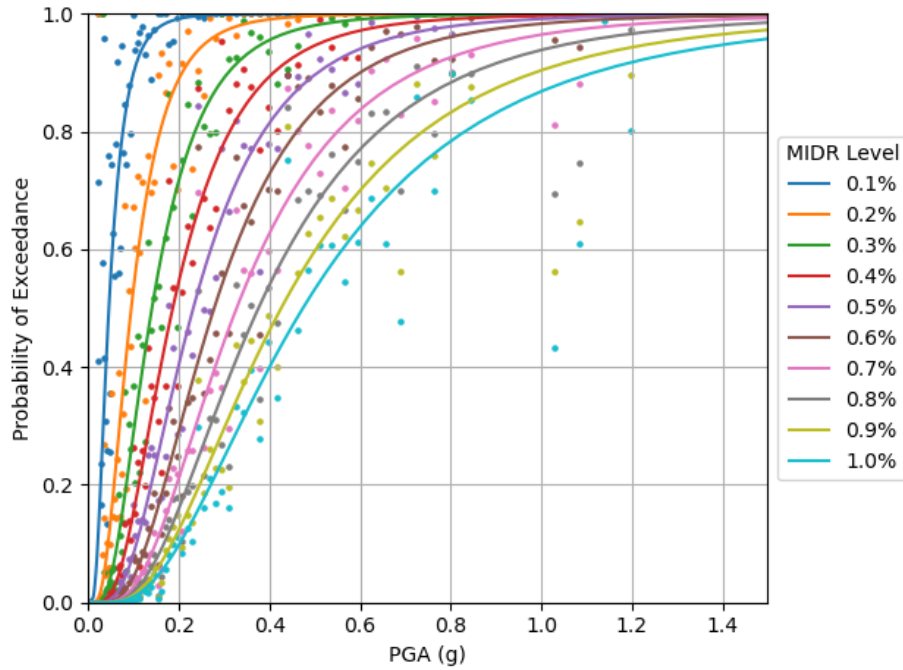


Figure 4.36. Fragility curves derived through log-normal distribution for PGA.

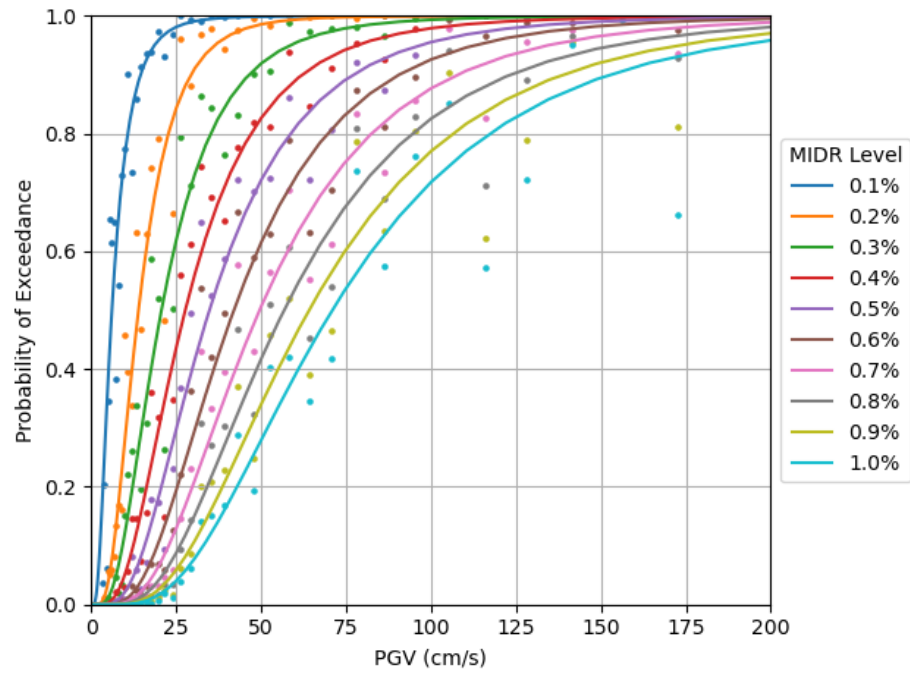


Figure 4.37. Fragility curves derived through log-normal distribution for PGV.

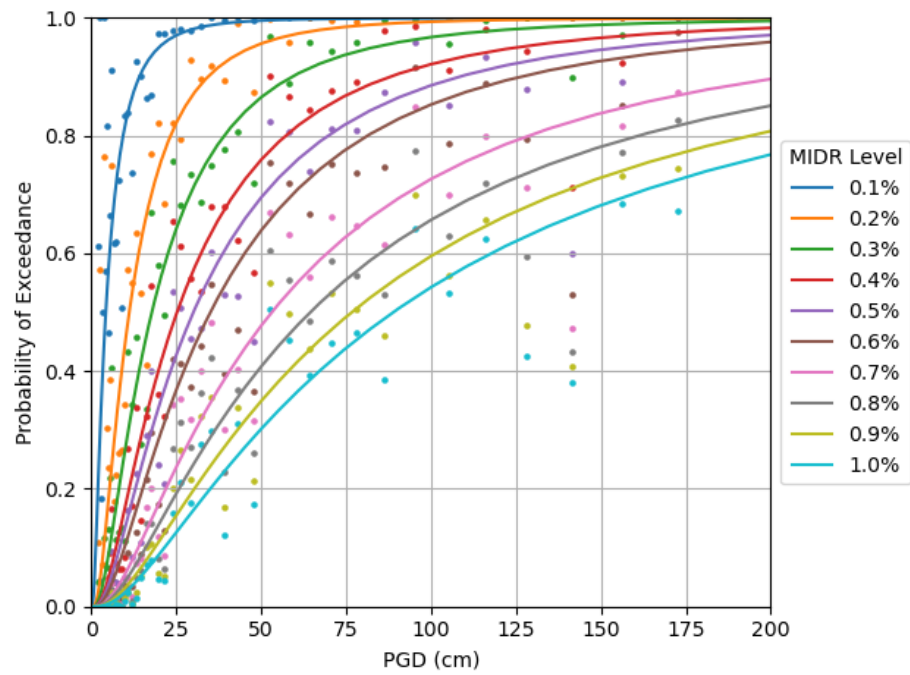


Figure 4.38. Fragility curves derived through log-normal distribution for PGD.

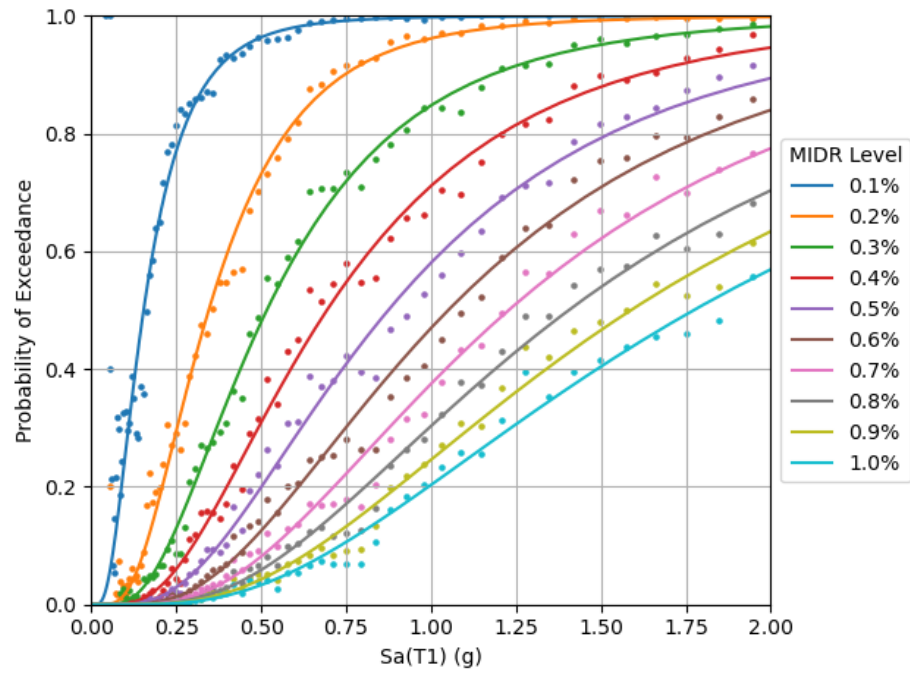


Figure 4.39. Fragility curves derived through log-normal distribution for $S_a(T_1)$.

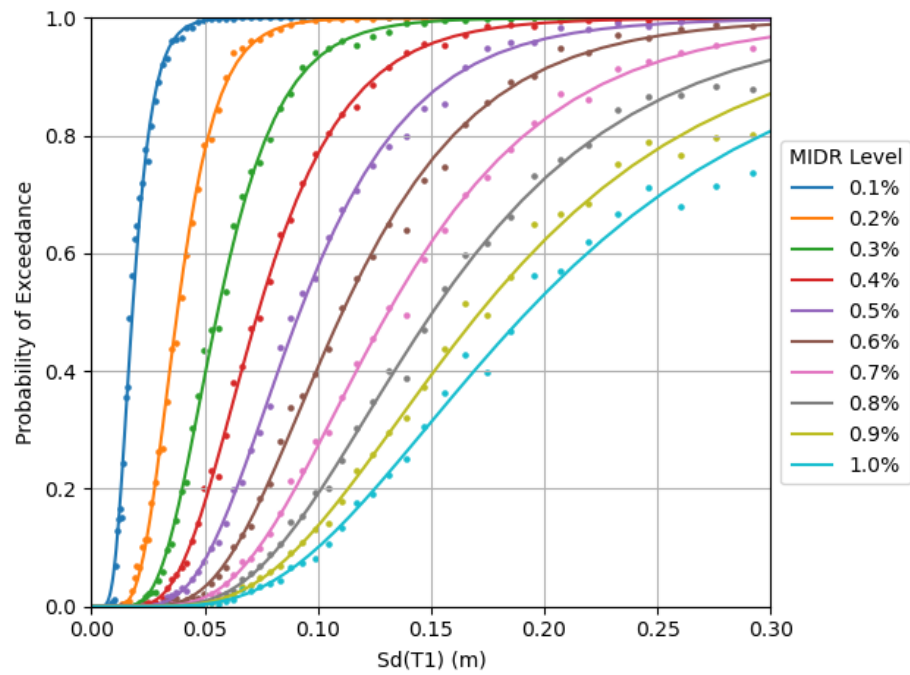


Figure 4.40. Fragility curves derived through log-normal distribution for $S_d(T_1)$.

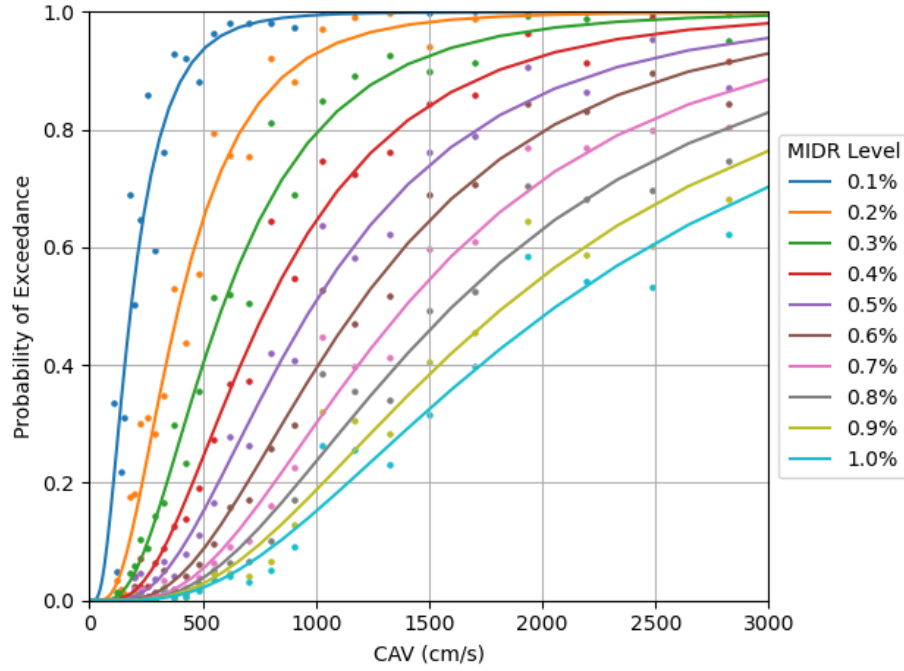


Figure 4.41. Fragility curves derived through log-normal distribution for CAV.

4.3. Comparison of Fragility Curves Derived from ML and Log-normal Distribution

Analytical fragility curves can differ from method to method, even if the utilized dataset is the same. However, the comparison of methods is necessary to get a general idea of the correct derivation of fragility curves in a logical framework. In this section, the comparison of the fragility curves derived from Machine Learning algorithms and fitting log-normal distribution with the maximum likelihood estimation method are illustrated in Figures between 4.42 and 4.71. In these figures, just the fragility curves of particular MIDR levels are presented for simple visualization.

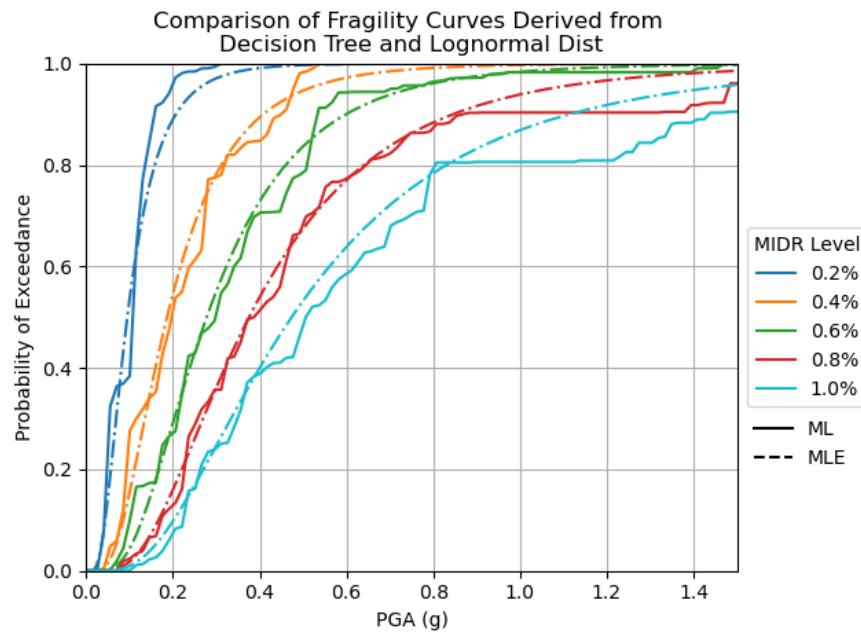


Figure 4.42. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA.

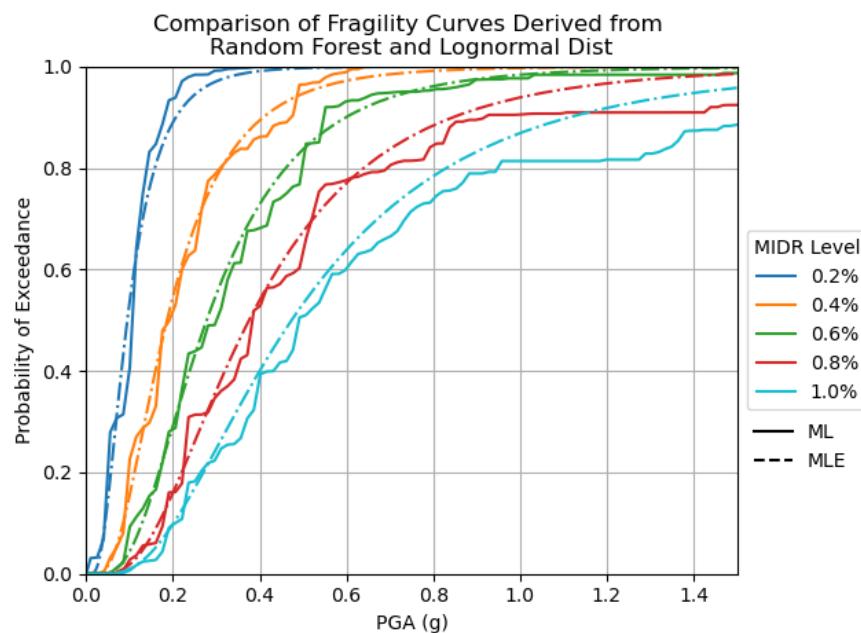


Figure 4.43. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA.

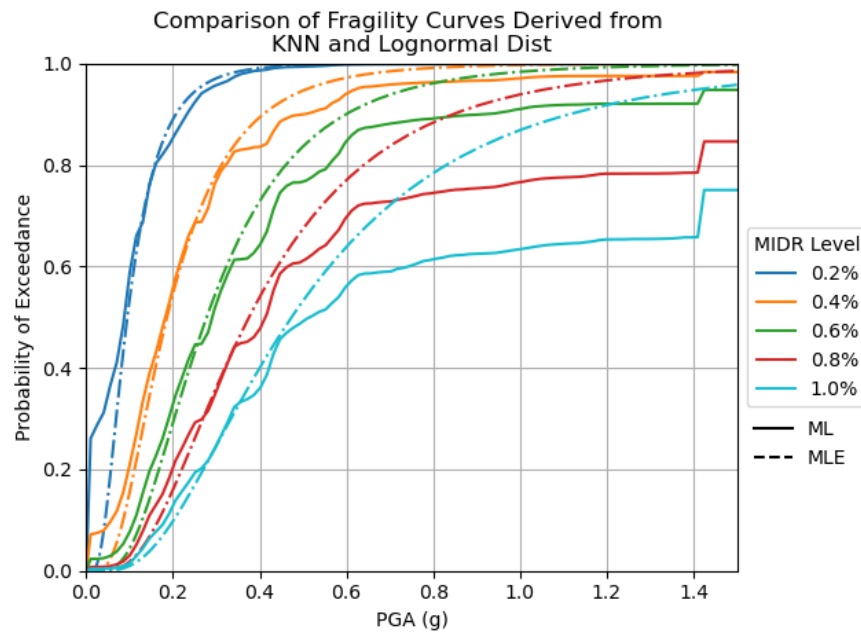


Figure 4.44. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA.

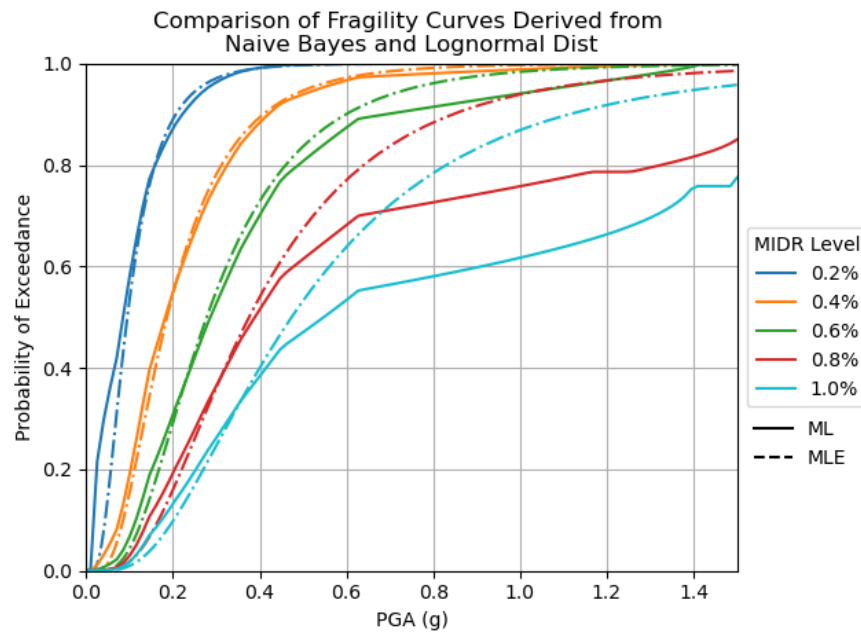


Figure 4.45. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA.

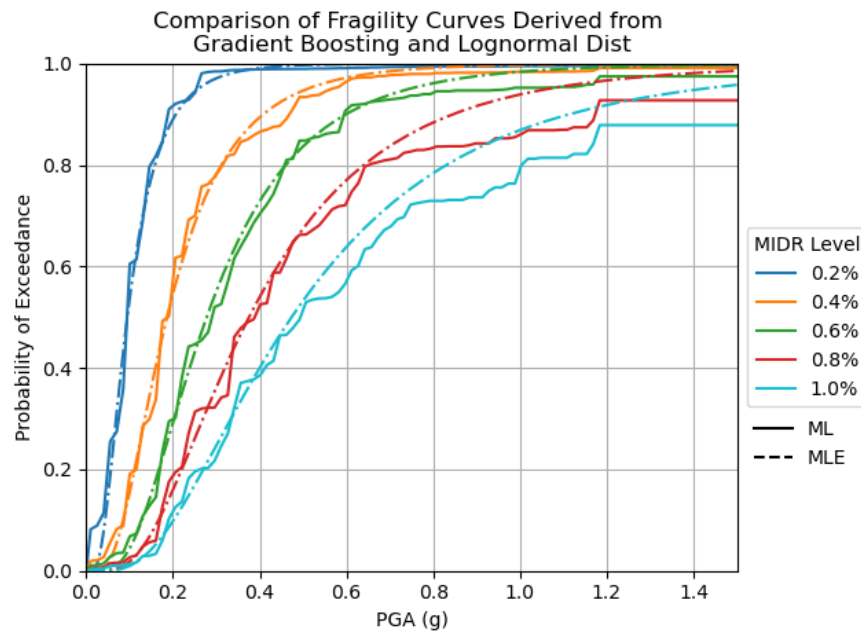


Figure 4.46. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGA.

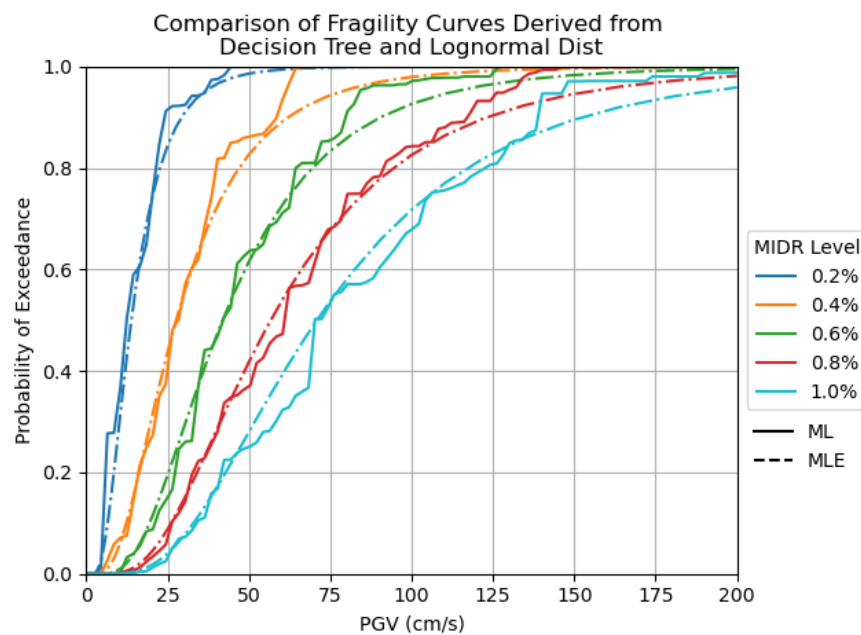


Figure 4.47. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV.

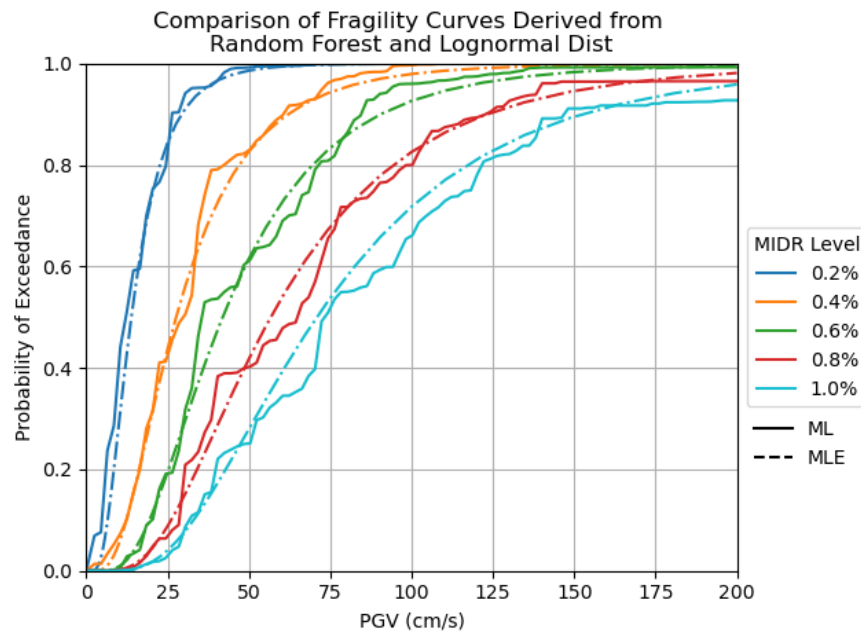


Figure 4.48. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV.

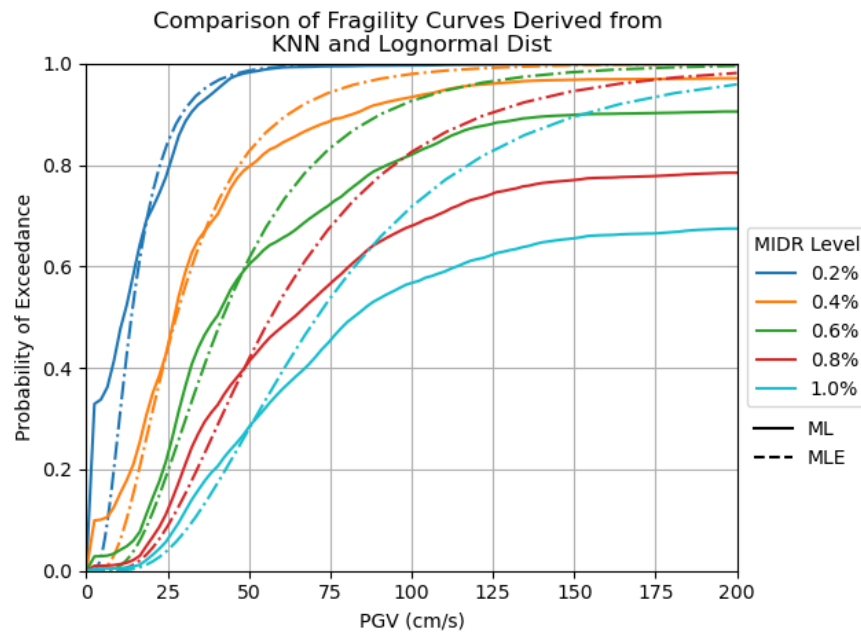


Figure 4.49. Comparison of the fragility curves derived from the K-Nearest Neighbor Algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV.

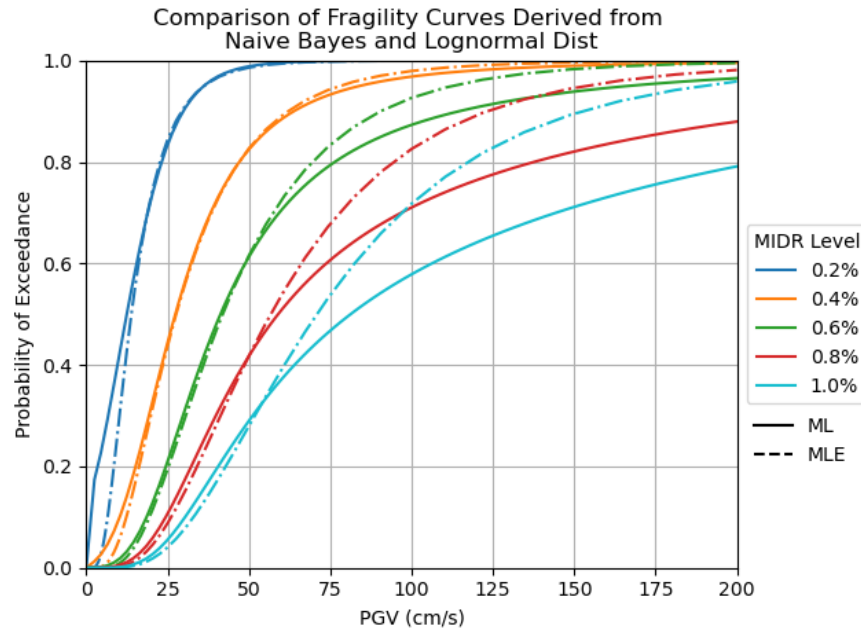


Figure 4.50. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV.

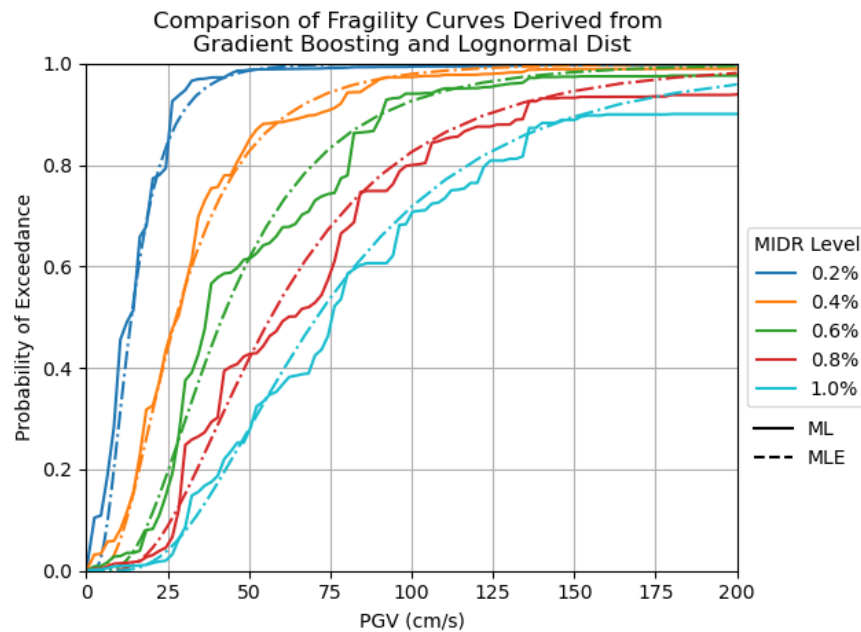


Figure 4.51. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGV.

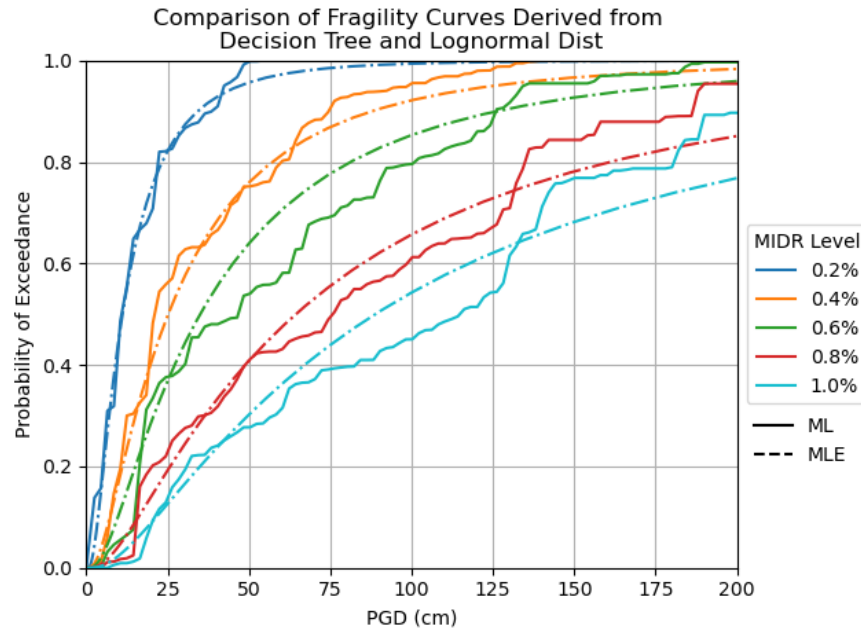


Figure 4.52. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.

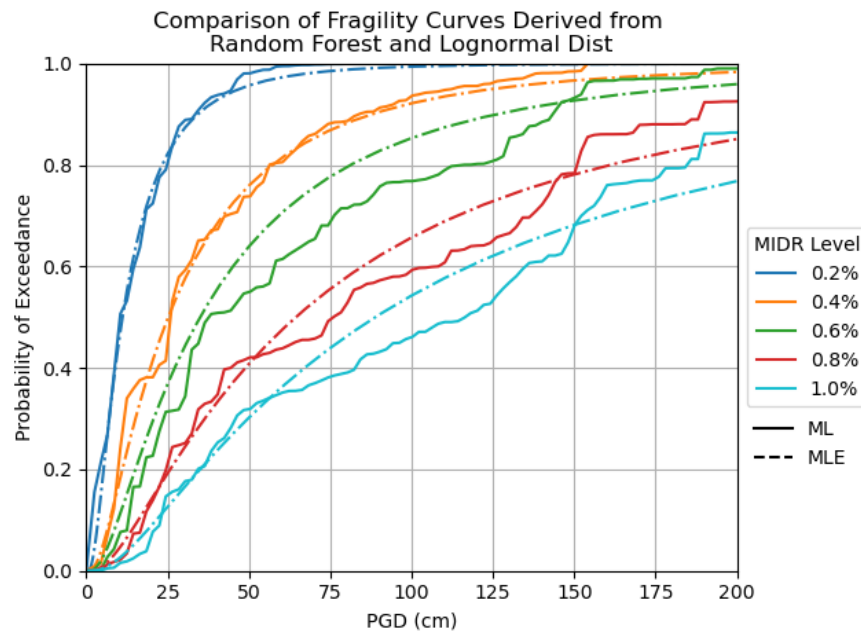


Figure 4.53. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.

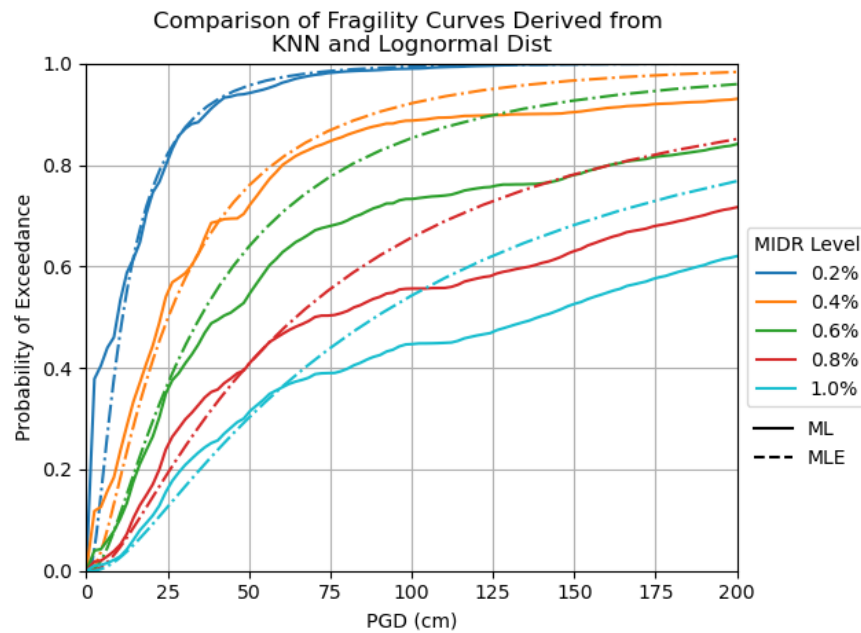


Figure 4.54. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.

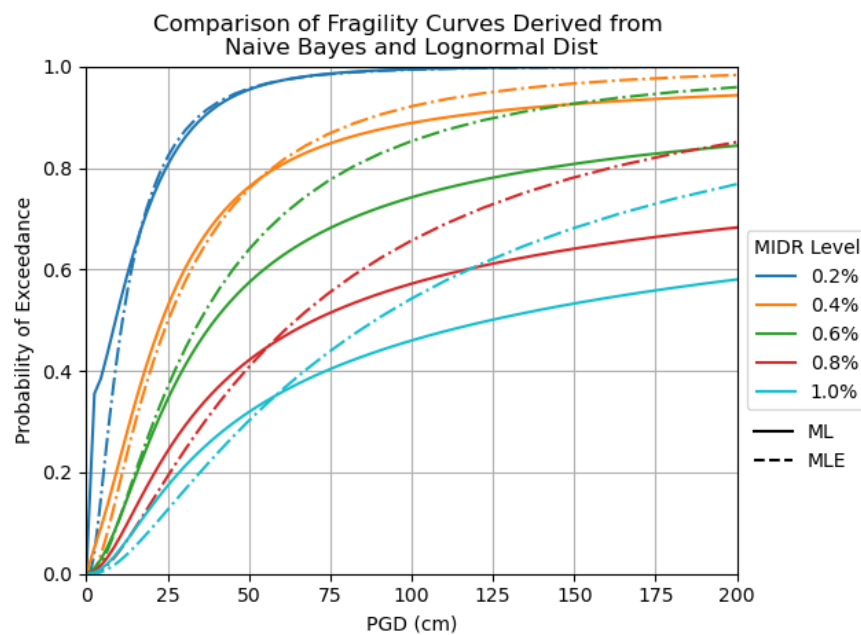


Figure 4.55. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.

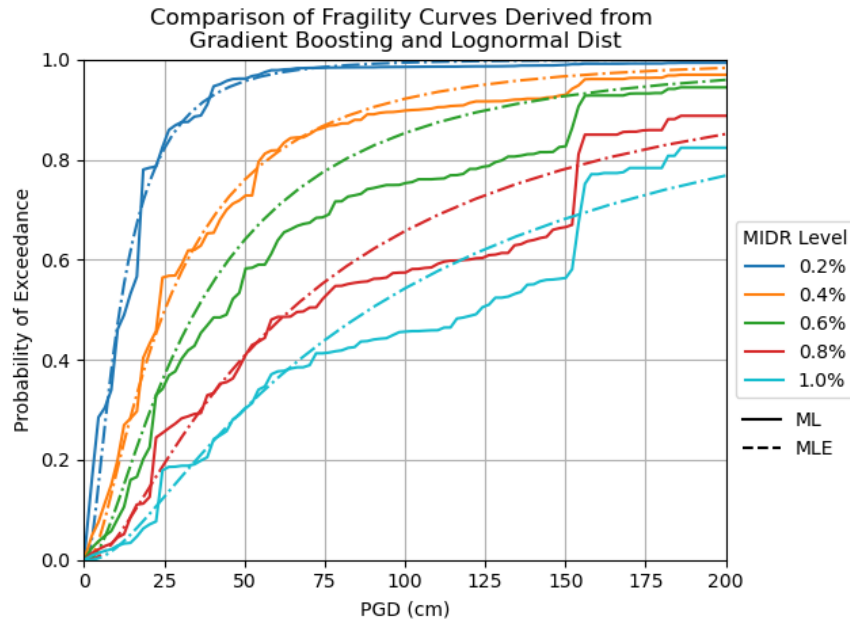


Figure 4.56. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of PGD.

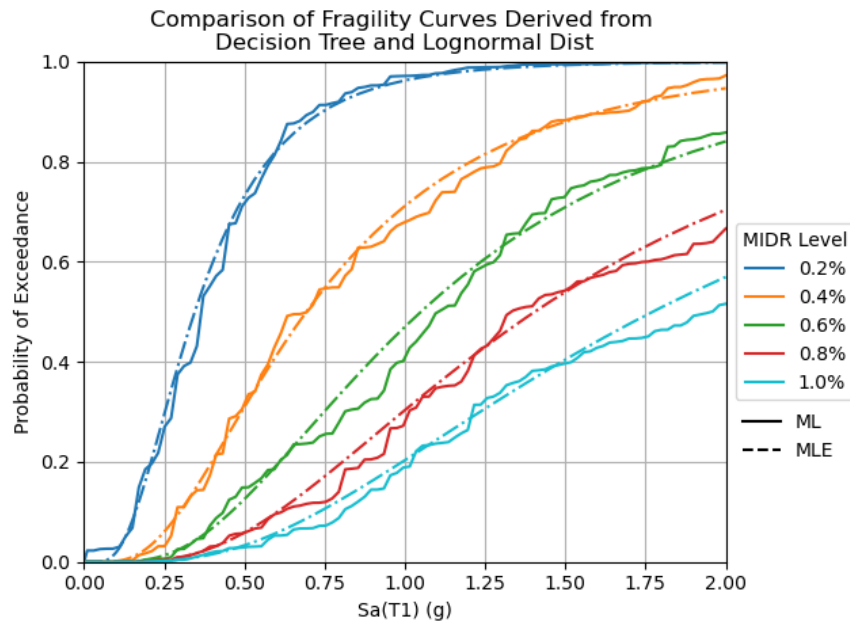


Figure 4.57. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$.

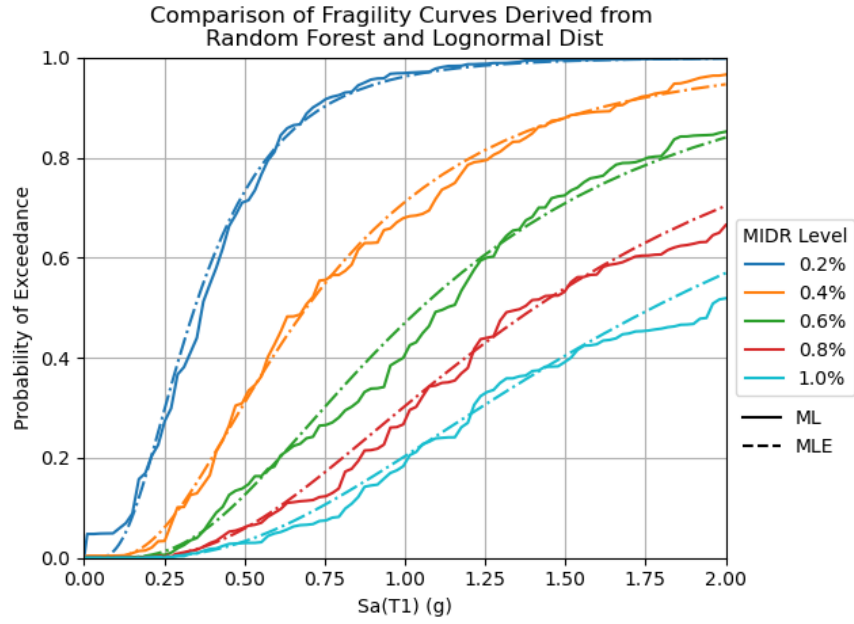


Figure 4.58. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$.

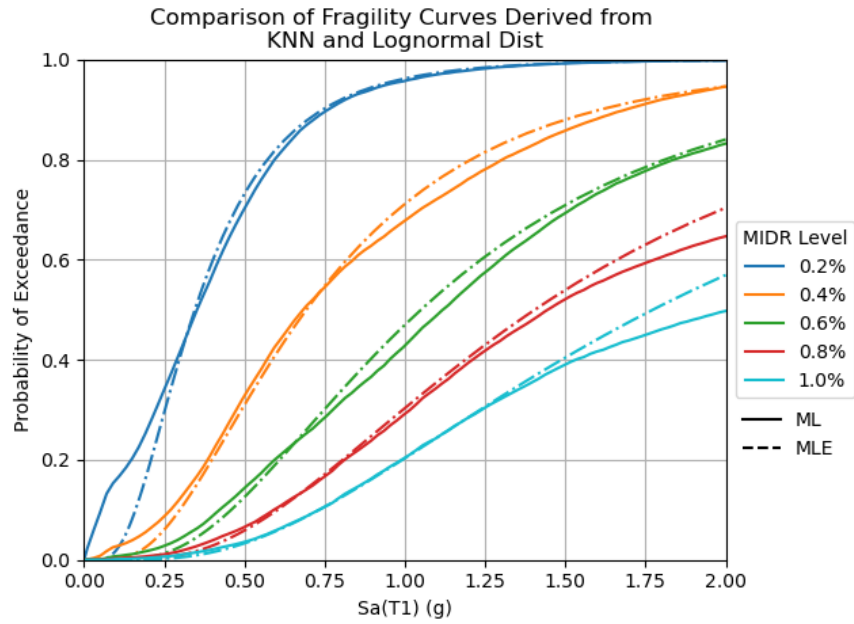


Figure 4.59. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$.

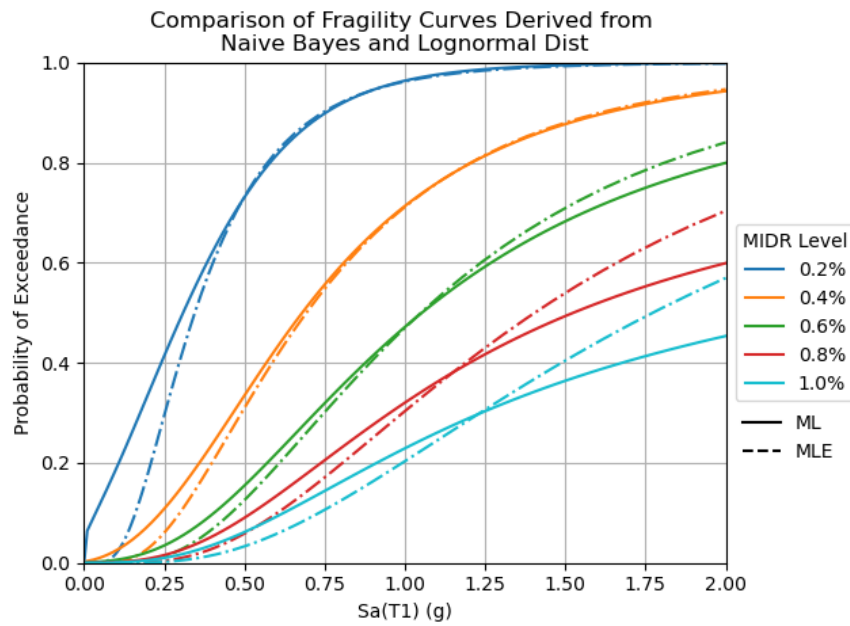


Figure 4.60. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$.

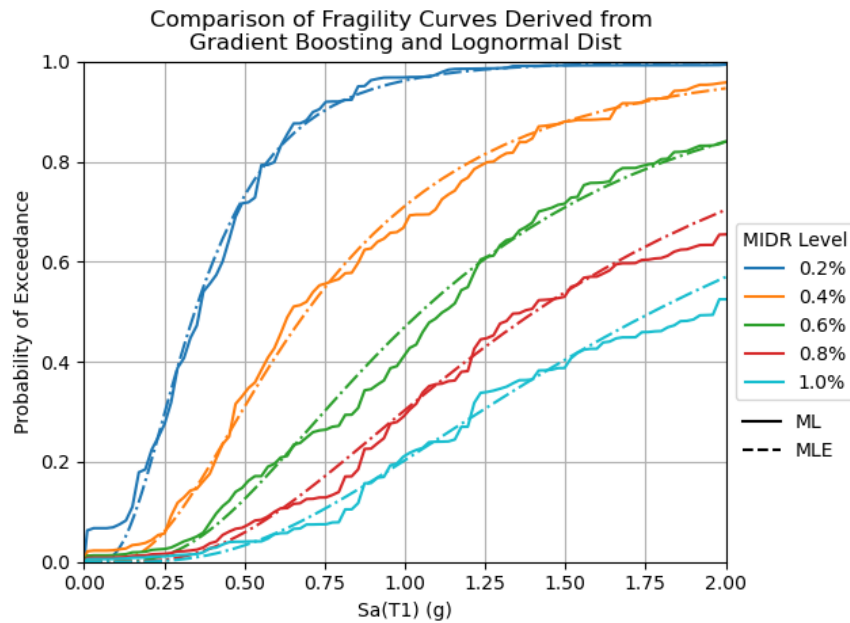


Figure 4.61. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_a(T_1)$.

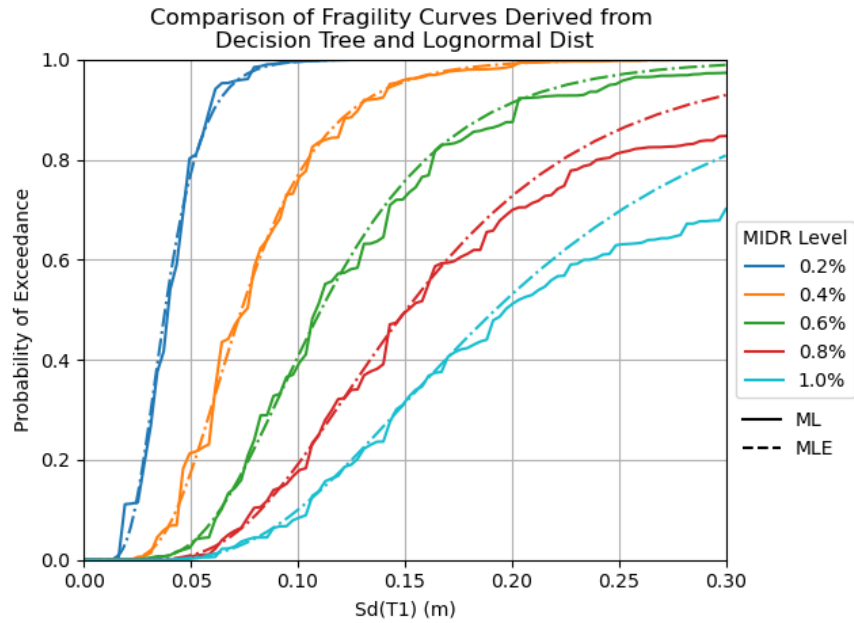


Figure 4.62. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$.

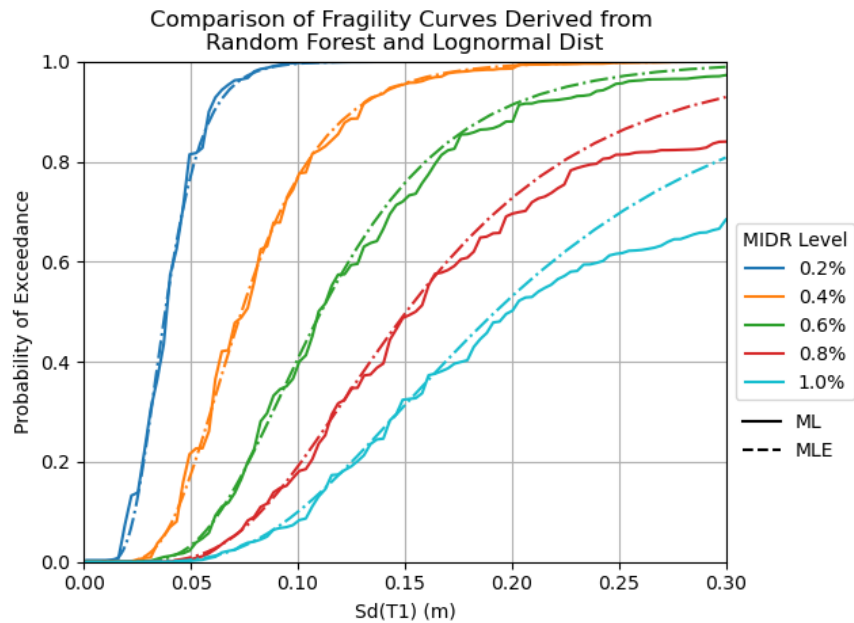


Figure 4.63. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$.

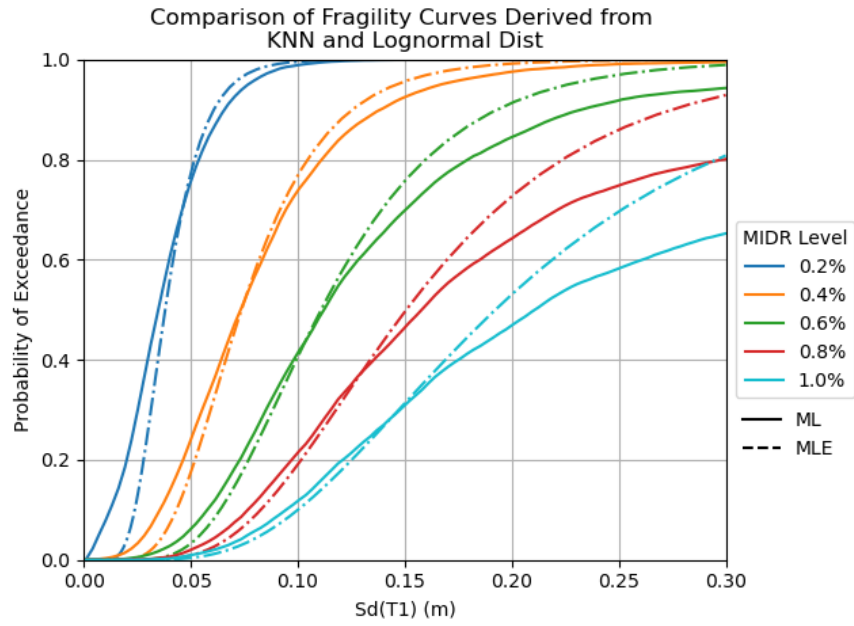


Figure 4.64. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$.

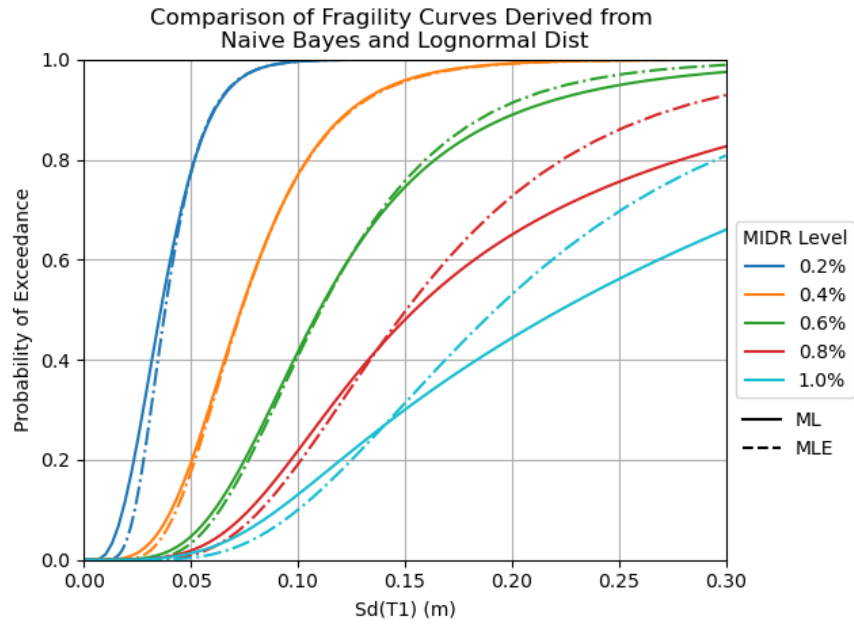


Figure 4.65. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$.

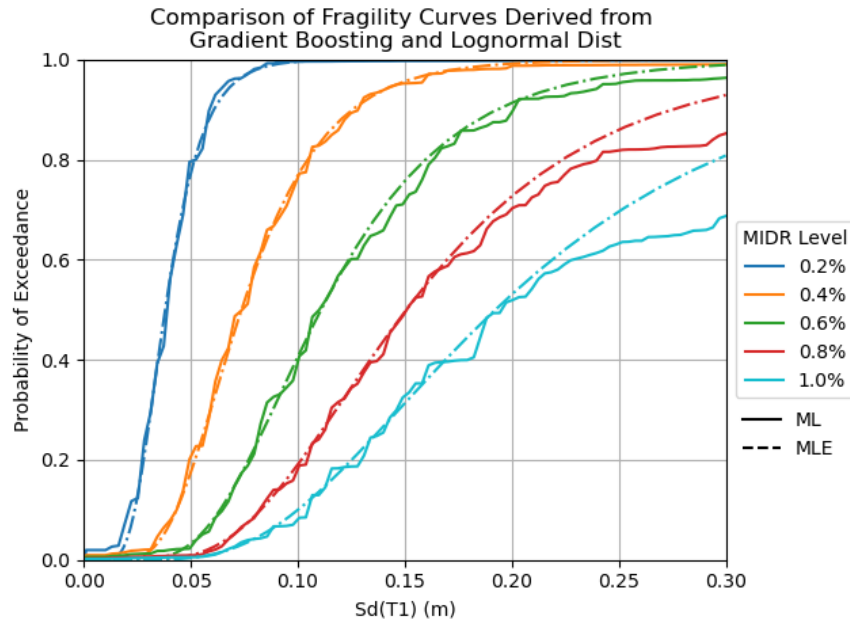


Figure 4.66. Comparison of the fragility curves derived from the Stochastic Gradient Boosting algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of $S_d(T_1)$.

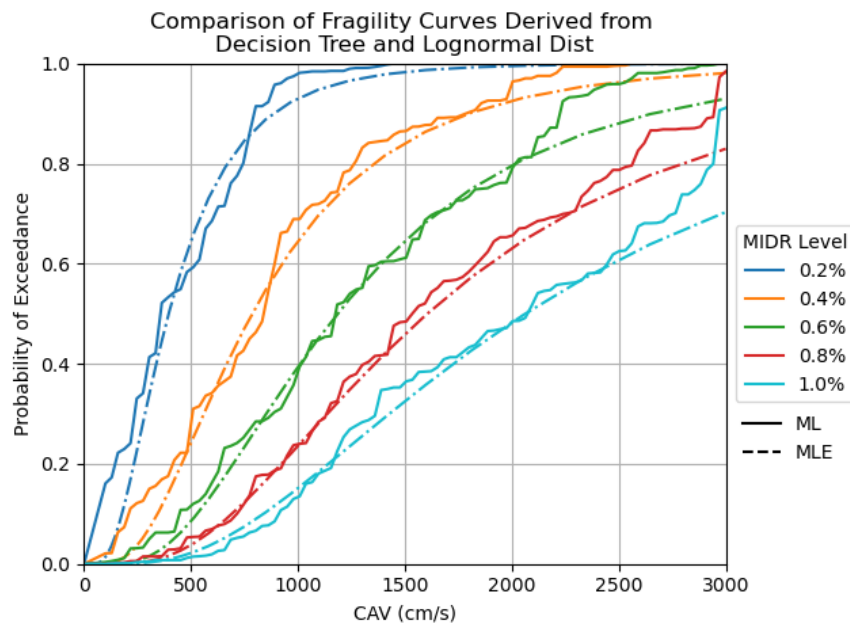


Figure 4.67. Comparison of the fragility curves derived from the Decision Tree algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV.

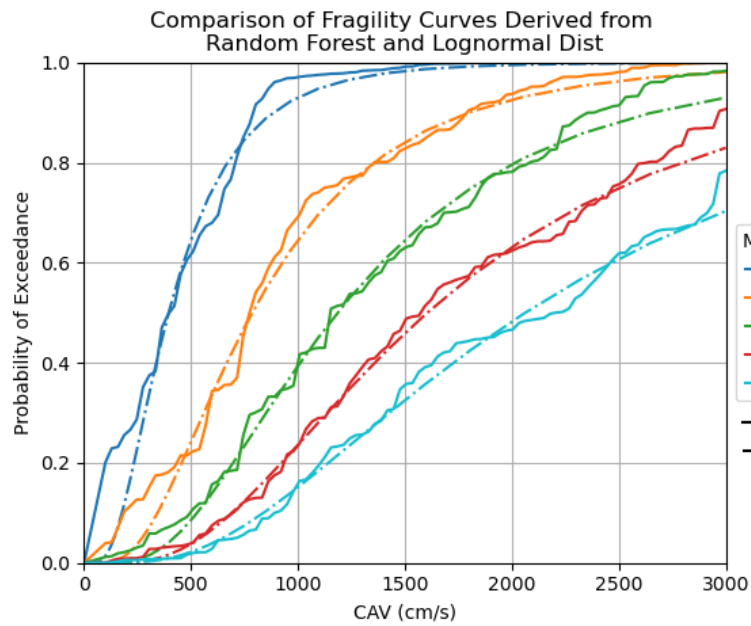


Figure 4.68. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV.

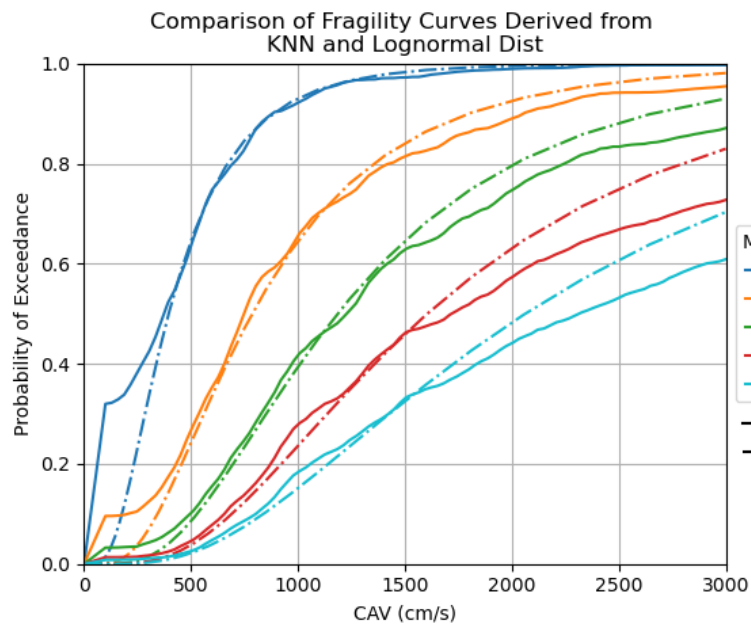


Figure 4.69. Comparison of the fragility curves derived from the K-Nearest Neighbor algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV.

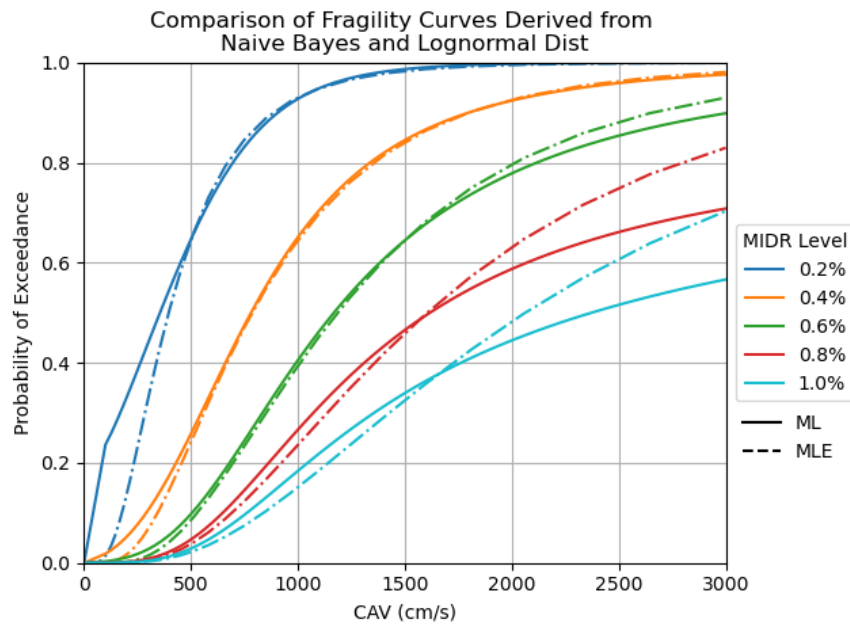


Figure 4.70. Comparison of the fragility curves derived from the Naïve Bayes algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV.

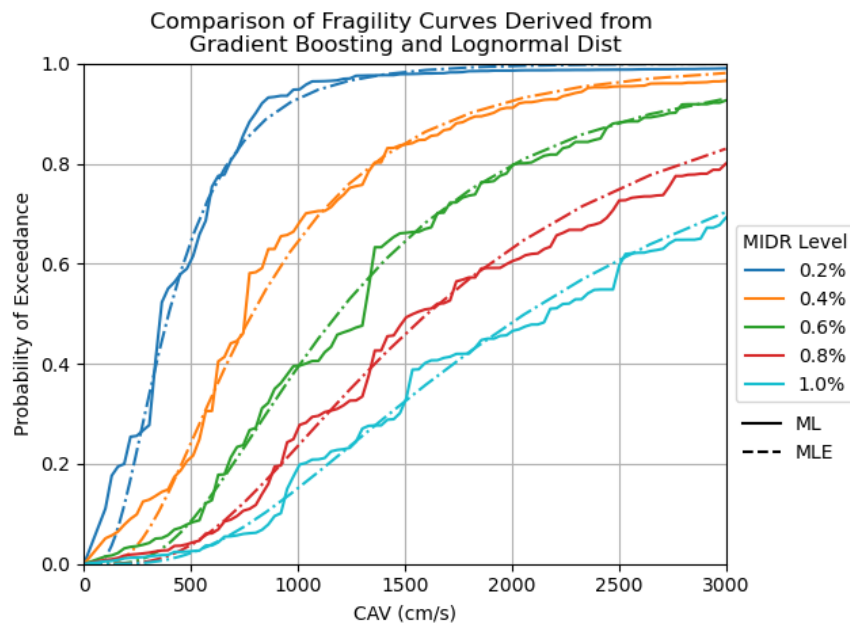


Figure 4.71. Comparison of the fragility curves derived from the Random Forest algorithm suggested in this study and fitting log-normal distribution via maximum likelihood method for a function of CAV.

Differences are noticeable at high values of the intensity measures. The main reason for these differences is that the sample weight is not taken into account when using the MLE method, while this is taken into account in machine learning algorithms. Therefore, it can be said that the fragility curves obtained by machine learning are more realistic. Nevertheless, the Random Forest algorithm trained using PGA as the intensity measure predicts the median MIDR level better, so it can be stated that this algorithm is the best algorithm for deriving fragility curves.

4.4. Comparison of Fragility Curves Derived in this Thesis with the Ones Used in Istanbul Loss Assessment Studies

There are numerous fragility curves in terms of using various data or methods. Therefore, it is crucial to compare the fragility curves to understand the reasons for differences. For this purpose, the fragility curves derived from the proposed method are compared with those presented in Erdik et al. (2003) for the Istanbul metropolitan area. Erdik et al. (2003) classified the buildings according to the type of the lateral load-resisting system (e.g. RC frame, masonry, etc), number of stories and the construction year, and provided log-normal fragility curves in terms of spectral displacement for each building class. Hence, fragility curves for mid-rise (5 to 8-story), reinforced concrete moment-resisting frame buildings constructed between 1980 and 2000 are utilized for comparison.

Table 4.3. MIDR thresholds for damage states of Mid-rise reinforced concrete buildings constructed after 1980 (Erdik et. al., 2003).

MIDR			
Slight Damage	Moderate Damage	Extensive Damage	Complete Damage
0.4%	1%	2%	4%

In this thesis, 1% MIDR is selected for complete damage; however, complete

damage is determined as 4% by Erdik et al. (2003). Therefore, 4% MIDR is not shown in figures 4.72, 4.73, 4.74, 4.75, and 4.76 in order not to be able to compare with proposed methods, which only cover the MIDR range between 0 and 1%. Moreover, the extensive damage by Erdik et al. (2003), which is 2%, is also not comparable; nevertheless, there is a good match between 1% of proposed methods and the extensive damage proposed by Erdik et al. (2003). Extensive damage is demonstrated in the figures to emphasize the similarity between these fragility curves.

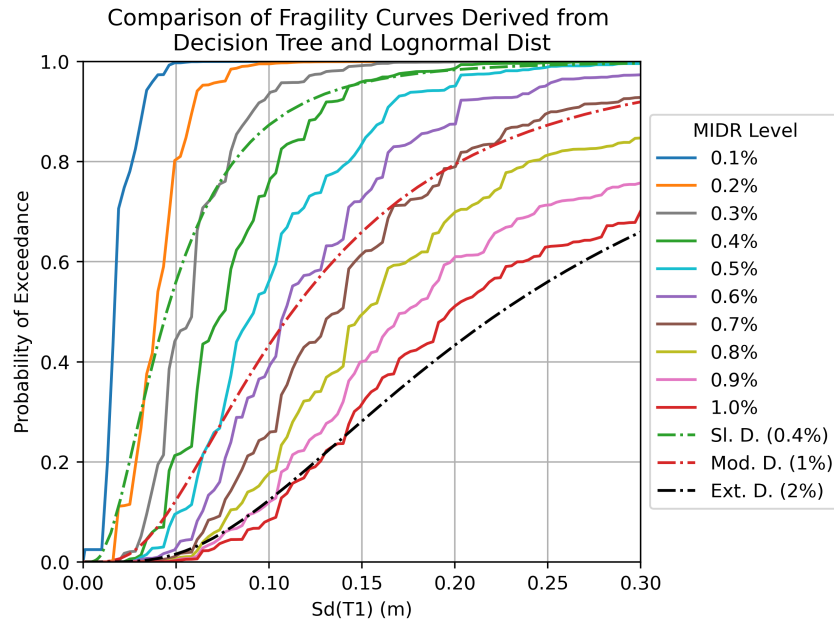


Figure 4.72. Comparison of the fragility curves with respect to spectral displacement derived from the Decision Tree suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines).

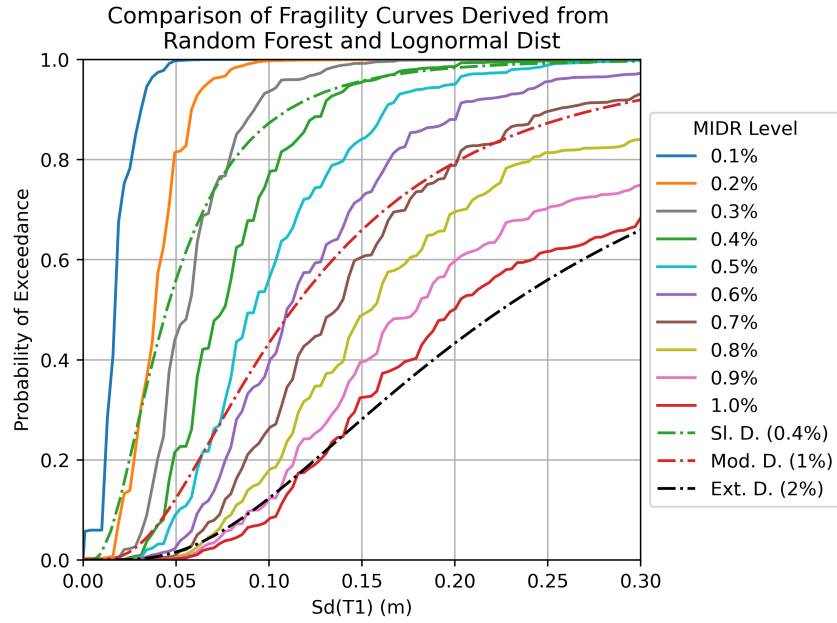


Figure 4.73. Comparison of the fragility curves with respect to spectral displacement derived from the Random Forest suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines).

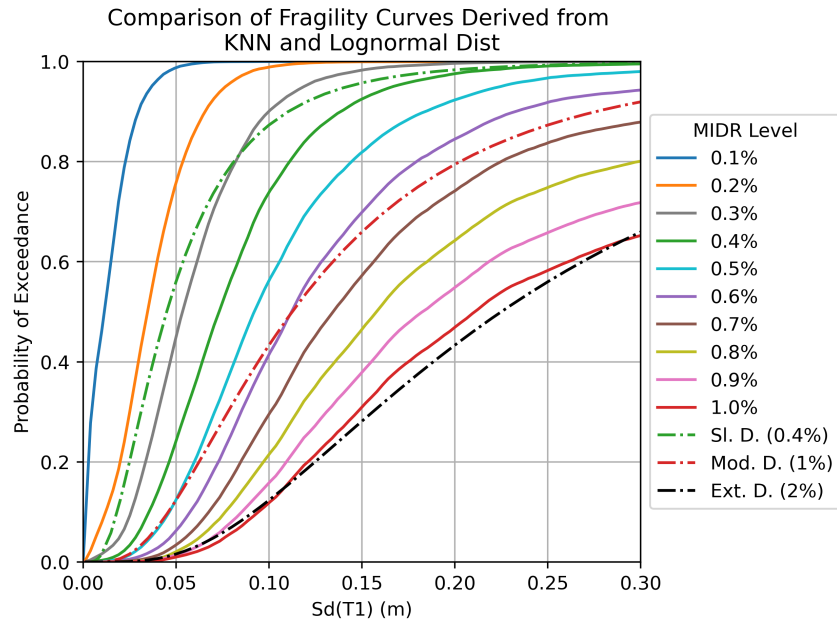


Figure 4.74. Comparison of the fragility curves with respect to spectral displacement derived from the K-Nearest Neighbor suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines).

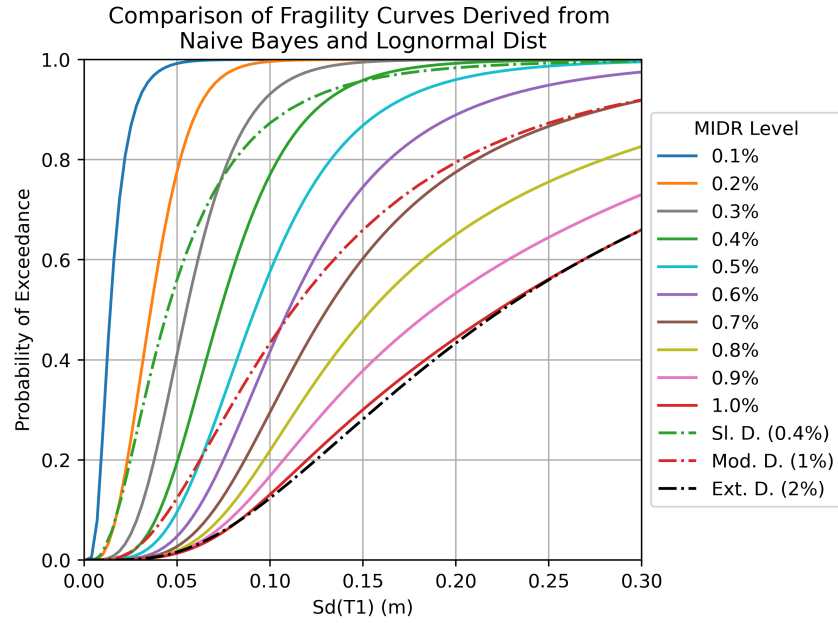


Figure 4.75. Comparison of the fragility curves with respect to spectral displacement derived from the Naïve Bayes suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines).

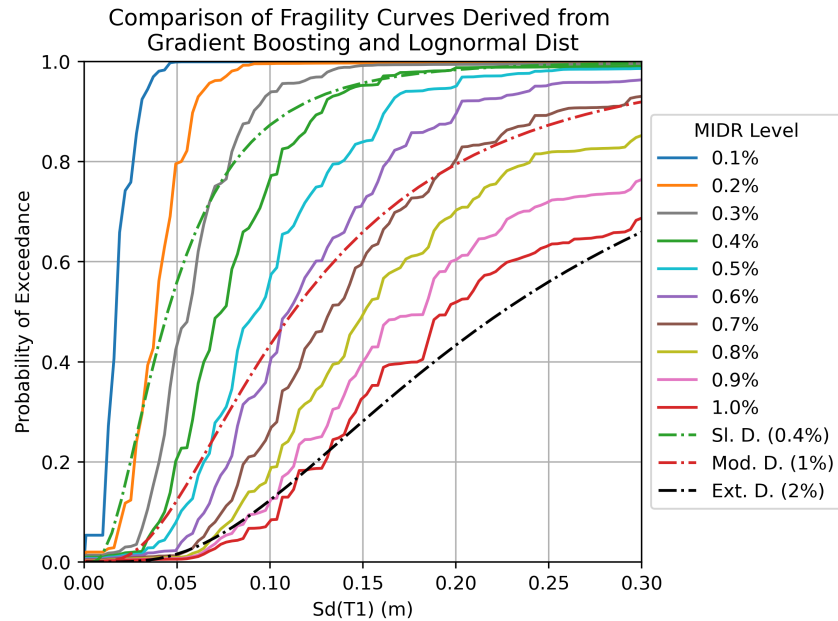


Figure 4.76. Comparison of the fragility curves with respect to spectral displacement derived from the Stochastic Gradient Boosting suggested in this thesis (solid lines) and mid-rise RC buildings constructed after 1980 in Erdik et al. (2003) (dashed lines).

The first observation is that slight damage shows tremendous similarity with the 0.2% MIDR at the low value of $S_d(T_1)$; however, when the $S_d(T_1)$ is increased, the fragility curves start close to the 0.4% MIDR, which is the same for slight damage proposed by Erdik et al. (2003). Around 0.15 m of $S_d(T_1)$, the difference between the fragility curves proposed in this thesis and Erdik et al. (2003) is almost invisible. Briefly, slight damage is more likely to be exceeded at low levels of $S_d(T_1)$ values when compared with 0.4% MIDR fragility curves, which means that slight damage is more vulnerable than the proposed methods in this thesis. In comparison, the exceedance probability of slight damage is extraordinarily close to 0.4% MIDR fragility curves at high levels of $S_d(T_1)$ values.

The second observation is that moderate damage, which is defined as 1% MIDR by Erdik et al. (2003), is not even close to the 1% MIDR fragility curves proposed in this thesis. The moderate damage fragility curve is covered by 0.4% and 0.7% fragility curves from low levels of $S_d(T_1)$ values to high levels of $S_d(T_1)$ values, respectively. So, the most visible observation is that the moderate damage fragility curve exceeds more than 1% MIDR fragility curves at any level of $S_d(T_1)$ values.

The last observation is that extensive damage, which is defined as 2% MIDR by Erdik et al. (2003), is remarkably similar to the 1% MIDR fragility curves proposed in this thesis.

As a result, these differences may be due to different definitions of structure characteristics or to the selection of strong ground motion records used both when calculating intensity measures and when performing the NLTH analysis used for the MIDR calculations corresponding to intensity measures.

5. CONCLUSION

In this thesis, generic buildings that represent the predominant building typology in the Marmara region, where many studies have shown a high probability of a severe earthquake occurrence within 30 years, are taken into consideration. It is aimed to comparatively apply different machine learning algorithms for the development of fragility functions for mid-rise RC buildings constructed between 1980 and 2000, which constitute the majority of the entire building stock and are deemed to be more vulnerable to strong ground shaking. Finite element models of the study buildings are elaborated in 2D based on the structural, material, and geometrical characteristics obtained from the available literature. The characteristics of the buildings, which are varied, are the number of stories, story height, commercial use of the first floor, number of spans, span length, column dimensions, and concrete compressive strength in the scope of this thesis. As a result, 1,080 representative buildings are modeled for investigation of the structural response under the earthquake records. Non-linear time history analysis is conducted to determine the engineering demand parameter of a representative building. As an engineering demand parameter, the maximum inter-story drift ratio (MIDR) is taken into consideration.

Strong ground motion records are selected through the conditional mean spectrum method. Each representative model is classified according to its first natural vibration period. For each class of the first natural vibration period, 11 ground motion records are selected. This ground motion records dataset is a combination of CyberShake and AFAD databases. 6 out of 11 are selected from the CyberShake database, while the rest of them are selected from earthquake records that happened in Turkey. This process is carried out for each first natural vibration period class. Additionally, the uncertainty in soil conditions is also considered in this thesis. Three soil conditions, which are stiff, moderate, and soft soil (ZB, ZC, ZD), are considered while selecting the ground motion records. A total of 330 earthquake records were selected. As a result, 35,640 non-linear time history analyses were conducted, and MIDR was calculated for each analysis.

The MIDR is divided into 10 groups, linearly spaced between 0% and 1%. 1% is accepted as a border of complete damage. The results obtained from non-linear time history analysis tend to complete damage, causing the generation of an imbalanced dataset. Therefore, selected ground motions are downscaled with a scale factor. This process is performed by utilizing 2 different scale factors. In the end, the raw dataset was created with 106,920 analysis results. The ground motion intensity measures considered in the derivation of fragility functions are peak ground acceleration (PGA), peak ground velocity (PGV), peak ground displacement (PGD), spectral acceleration $[S_a(T_1)]$, spectral displacement $[S_d(T_1)]$, and cumulative absolute velocity (CAV).

The raw dataset was prepared for machine learning (ML) algorithm with pre-processing of the features (building characteristics and ground motion intensity measure). Then, 5 supervised classification algorithms are applied to this processed dataset. These machine learning algorithms are Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN), Naïve Bayes (NB), and Stochastic Gradient Boosting (SGB). In addition, the hyperparameter tuning is carried out to obtain the best-fitted hyperparameters for each algorithm.

In conclusion, the fragility curves are derived with these ML algorithms and compared to the ones generated by fitting log-normal distribution with the maximum likelihood estimation (MLE) method, which is one of the conventional methods, and those derived by Erdik et al. (2003) for Istanbul Metropolitan area.

5.1. Summary of Findings

The first finding of the thesis is that the Random Forest algorithm is the best prediction algorithm among the others, with a F_1 score, which is one of the most significant metrics to interpret the accuracy of the model, above 78% when PGA, PGV, PGD, and CAV are used as intensity measures in the dataset.

The second observation from the results is that the Decision Tree is the second-best prediction algorithm with an F_1 score above 75%. The high F_1 score is obtained in the Decision Tree, which is similar to the Random Forest algorithm when PGA, PGV, PGD, and CAV are used in the dataset.

It is observed that tree-based algorithms with if-else logic are successful in predicting the median value. These tree-based algorithms are the Decision Tree, Random Forest, and Stochastic Gradient Boosting algorithms.

The Stochastic Gradient Boosting method is the best-performing method to estimate the complete damage buildings with a recall metric above 93% for all intensity measures.

Each building can be assessed through these ML algorithms without any additional process or without analyzing this building with non-linear analysis under real earthquake records, which requires intense structural modeling and analysis efforts and considerable time allocation. The results show that the probability of being at MIDR level when encountered with a specific intensity value is sufficiently predicted with the ML algorithms utilized in the scope of this thesis. In addition, with this method, when a building characteristic cannot be obtained due to lack of information, it can be predicted with the ML algorithm by utilizing the distribution of that characteristic of the buildings in the region where that building is located. This is an advantage of this method over conventional methods.

Moreover, the fragility curves can be derived through the methods proposed in the thesis. The fragility curves derived from tree-based algorithms are represented as a step function. The reason for that is that these algorithms are based on if-else statements, so there is no continuous function that separates intensity measures. In contrast, the KNN and Naïve Bayes algorithms make predictions through intensity measure as a function of a continuous line. This continuous line can be expressed as a normal cumulative distribution function as in the Naïve Bayes algorithm, a polynomial line, or a straight line.

The fragility curves derived from the proposed methods and the conventional method, which is fitting log-normal distribution, are compared with each other. The results show that fragility curves derived from the proposed methods are satisfactorily sufficient. There are small differences between the conventional method and the proposed methods at high values of the intensity measures. The reason is that there is not enough sample at a high value of ground motion intensity measure. The conventional method does not consider the sample weight; on the other hand, ML algorithms take the sample weight into account since these algorithms make data-specific predictions.

Another important observation can be made for the comparison of fragility curves proposed by Erdik et al. (2003) and in this thesis. At low $S_d(T_1)$ values, the probability of exceeding the slight damage fragility curve proposed by Erdik et al. (2003) is higher than the probability of exceeding the 0.4% fragility curves produced in this thesis, while at high $S_d(T_1)$ values, the probabilities of exceedance almost coincide. In addition, when the MIDR fragility curves derived in this thesis with 0.1% differences are taken into account, the slight damage fragility curve derived in Erdik et al. (2003) falls between 0.2% and 0.4% MIDR fragility curves. Moreover, the moderate damage fragility curve generated by Erdik et al. (2003), considering a 1% MIDR, indicates a higher probability of exceedance for each $S_d(T_1)$ value compared to the 1% MIDR fragility curve produced within the scope of this thesis. However, the moderate damage fragility curve can be covered by the fragility curves between 0.4% MIDR and 0.7% MIDR fragility curves. Furthermore, the extensive damage fragility curve, which is

defined as 2% MIDR in Erdik et al. (2003), is quite similar to the 1% MIDR fragility curves derived from this thesis. In conclusion, the differences noted may arise from diverse interpretations of structural characteristics, the selection of seismic records utilized in intensity measure calculations and NLTH analyses for corresponding MIDR calculations, or the utilized method itself.

The last observation, but not least, is the performance of ML algorithms when $S_a(T_1)$ and $S_d(T_1)$ used in the dataset as an intensity measure is lower than the other intensity measures. The reason for this observation is that there is an interdependency between $S_a(T_1)$, $S_d(T_1)$, and the selected ground motion due to the utilization of the conditional mean spectrum while selecting ground motion records. Because of this reliance, these algorithms may offer less accurate predictions.

As a result, well-trained machine learning algorithms can accelerate the process of assessing buildings, reduce computational costs, and derive trustworthy fragility curves. Additionally, they eliminate the limitations and allow us to produce faster solutions.

5.2. Recommendation for Further Studies

This thesis is confined to mid-rise buildings constructed between 1980 and 2000. This method can be applied to all types of buildings constructed by considering different earthquake building codes. In addition, this approach will yield more realistic results when the dataset is generated from the nonlinear time history analysis of 3D buildings. Another essential improvement will be to take into account the variability of more building characteristics. These further studies help to improve the reliability of the buildings.

Another crucial improvement can be made in the selection of strong ground motion records. This improvement would be the selection of earthquakes by disaggregation method considering certain exceedance probability of the intensity measures according

to the probabilistic seismic hazard analysis at a location. Moreover, an improved study can be conducted by making selections at different locations, taking into account the far-field and near-fault locations.

Furthermore, the proposed method can be enhanced by adding real observational data of damaged and undamaged buildings obtained after an earthquake. This helps to calibrate the analytical data with observational data.

Finally, other artificial intelligence methods, such as deep learning or another type of machine learning algorithm, can be used to predict damage or derive fragility curves. In addition, selected two or more algorithms can be aggregated to obtain an ensemble algorithm to predict more accurately.

REFERENCES

- ABYYHY., 1975, “Regulation on Buildings to be Built in Disaster Areas, Ministry of Construction and Settlement”. (In Turkish: Afet Bolgelerinde Yapılacak Yapılar Hakkında Yonetmelik, İmar ve İskan Bakanlığı) Ankara, Türkiye.
- AFAD-TADAS. Earthquake Department of the Disaster and Emergency Management Presidency, 2020, Ankara, Türkiye, <https://tadas.afad.gov.tr>, accessed on Jan 11, 2023.
- Ahmad, N., H. Crowley, and R. Pinho, 2011, “Analytical fragility functions for reinforced concrete and masonry buildings aggregates of Euro Mediterranean regions – UPAV methodology”, Internal report, Syner-G Project, 2009/2012.
- Akkar, S., T. E. Azak, T. Can, U. Ceken, M. Demircioglu, T. Duman, M. Erdik, S. Ergintav, F. Kadirioglu, D. Kalafat, O. Kale, R. Kartal, K. Kekovali, T. Kilic, S. Ozalp, S. Poyraz, K. Sesetyan, S. Tekin, A. Yakut, and O. Fercan, 2018, “Evolution of seismic hazard maps in Turkey”, *Bulletin of Earthquake Engineering*, Vol. 16, pp. 3197-3228
- Akyuz, S., and M. Uyan, 1992, “A study on the concrete steel bars used in Turkey”. *Tech J Turk Assemb Civil Eng*, Vol. 35, pp. 497–508, [in Turkish].
- Alpaydin, E., 2014, *Introduction to Machine Learning*, Third Edition, MIT Press, Massachusetts.
- Atakan, K., A. Ojeda, M. Meghraoui, A. Barka, M. Erdik, and A. Bodare, 2002, “Seismic hazard in Istanbul following the August 17, 1999 Izmit and November 12, 1999 Düzce earthquakes”, *Bulletin of the Seismological Society of America*, Vol. 92, No. 1, 466–482.

- Atici, A. T., O. Ulku, and U. Hancilar, 2023, “Development of a seismic damage prediction model by using machine learning algorithms with an artificial dataset”, *Society for Earthquake and Civil Engineering Dynamics (SECED) 2023 Conference*, Cambridge, United Kingdom.
- Azak, T. E., B. Ö. Ay, and S. Akkar, 2014, “A statistical study on geometrical properties of Turkish reinforced concrete building stock”, *2nd European Conference on Earthquake Engineering and Seismology*, Istanbul, Türkiye.
- Baker, J. W., 2011, “Conditional mean spectrum: Tool for ground-motion selection”, *Journal of Structural Engineering*, Vol. 137, pp. 322–331.
- Baker, J. W., 2015, “Efficient Analytical Fragility Function Fitting Using Dynamic Structural Analysis”, *Earthquake Spectra*, Vol. 31, No. 1, pp. 579–599.
- Baker, J. W., B. Bradley, and P. Stafford, 2021, *Seismic Hazard and Risk Analysis*, Cambridge University Press, Cambridge
- Baker, J. W., C. Goulet, N. Luco, S. Rezaeian, and G. Teng, 2021, “A Subset of CyberShake Ground Motion Time Series for Response History Analysis.” *Earthquake Spectra*, Vol. 37, No. 2, pp. 1162-1176
- Baker, J. W., and C. Lee, 2018, “An Improved Algorithm for Selecting Ground Motions to Match a Conditional Spectrum”, *Journal of Earthquake Engineering*, Vol. 22, No. 4, pp. 708–723.
- Bal, E.I., H. Crowley, R. Pinho, and F.G. Gülay, 2008, “Detailed assessment of structural characteristics of Turkish RC building stock for loss assessment models”, *Soil Dynamics and Earthquake Engineering*, Vol. 28, No. 10–11, pp. 914-932.

- Box, G.E.P., and D.R. Cox, 1964, “An Analysis of Transformations”, *Journal of the Royal Statistical Society B*, Vol. 26, pp. 211-252.
- Breiman, L., 2001, “Random Forests”, *Machine Learning*, Vol. 45, pp. 5–32.
- Burgos, R. B., and L. E. Silva, 2023, “Evaluation of the P- (P-Delta) effect in columns and frames using the two-cycle method based on the solution of the beam-column differential equation”, *MethodsX*, Vol. 11, pp. 102248
- Cagnan, Z. and S. Akkar, 2010, “A Local Ground-Motion Predictive Model for Turkey, and Its Comparison with Other Regional and Global Ground-Motion Models”, *Bulletin of the Seismological Society of America*, Vol. 100, pp. 2978-2995.
- Chiou B. S. -J., and R. R. Youngs, 2014, “Update of the Chiou and Youngs NGA Model for the Average Horizontal Component of Peak Ground Motion and Response Spectra”, *Earthquake Spectra*, Vol. 30, No. 3, pp. 1117-1153.
- Chopra A. K., 2013, *Dynamic of Structures: Theory and Applications to Earthquake Engineering*, Fourth Edition, Pearson Education, NJ.
- Crowley, H., and R. Pinho, 2011, “Analytical Fragility Functions for Reinforced Concrete and Masonry Structures and Structures Aggregates of Euro-Mediterranean Regions - UPAV methodology”, Tech. rep.
- CSI, “SAP2000 Integrated Software for Structural Analysis and Design,” Computers and Structures Inc., Berkeley, California.
- DEZIM, 2020, “The project of updating probable earthquake losses estimation for Istanbul province”, Istanbul Metropolitan Municipality, Directorate of Earthquake and Ground Research.

- Erdik, M., M. Demircioglu, K. Sesetyan, E. Durukal, and B. Siyahi, 2004, “Earthquake hazard in Marmara Region, Turkey”, *Soil Dynamics and Earthquake Engineering*, Vol. 24, No.8, pp. 605–631.
- Erdik, M., N., Aydinoglu, Y. Fahjan, K. Sesetyan, M. Demircioglu, B. Siyahi, E. Durukal, C. Ozbey, Y. Biro, H. Akman, and O. Yuzugullu, 2003, “Earthquake risk assessment for Istanbul metropolitan area”, *Earthquake Engineering and Engineering Vibration*, Vol. 2, No. 1, pp. 1–23.
- Ghobarah, A., 2004, “On drift limits associated with different damage levels”, *International Workshop on Performance-Based Seismic Design: concepts and implementations*, Department of Civil Engineering, McMaster University.
- Gulerce, Z. and S. Ocak, 2013, “Probabilistic seismic hazard assessment of Eastern Marmara Region”, *Bulletin of Earthquake Engineering*, Vol. 11 No.5, pp. 1259–1277.
- Hancilar, U., C. Tuzun, C. Yenidogan, and M. Erdik, 2010, “ELER software -A new tool for urban earthquake loss assessment”, *Natural Hazards and Earth System Sciences*, Vol. 10, No. 12, pp. 2677–2696.
- Hubert-Ferrari, A., A. Barka, E. Jacques, S. Nalbant, B. Meyer, R. Armijo, P. Tapponnier and G. C. P. King, 2000, “Seismic hazard in the Marmara Sea following the 17 August 1999 Izmit earthquake”, *Nature*, Vol. 404, pp. 269-272.
- Ince, H., E. Toy, and M. Tolon, 2018, “Influence of Local Soil Conditions on the Structural Design and Associated Costs”, *International Journal of Engineering and Natural Sciences*, Vol. 1, No. 1, pp. 21-26.

- Jalayer, F., R. De Risi, and G. Manfredi, 2014, “Bayesian Cloud Analysis: efficient structural fragility assessment using linear regression”. *Bulletin of Earthquake Engineering*, Vol. 13, No. 4, pp. 1183–1203.
- Kalkan, E., P. Güllkan, N. Yilmaz, and M. Çelebi, 2009, “Reassessment of probabilistic seismic hazard in the Marmara Region”, *Bulletin of the Seismological Society of America*, Vol. 99, No. 4, pp. 2127–2146.
- Kaur, H., H. S. Pannu, and A. K. Malhi, 2020. “A systematic review on imbalanced data challenges in machine learning”, *ACM Computing Surveys*, Vol. 52, No.4, pp. 1–36.
- Kennedy, R. P., C. A. Cornell, R. D. Campbell, S. Kaplan, and H. F., Perla, 1980, “Probabilistic Seismic Safety Study of an Existing Nuclear Power Plant”, *Nuclear Engineering and Design*, Vol. 59, No. 2, pp. 315–338.
- Kennedy, R. P., and M. K., Ravindra, 1984, “Seismic Fragilities for Nuclear Power Plant Risk Studies”, *Nuclear Engineering and Design*, Vol. 79, No. 1, pp. 47–68.
- Kiani J., C. Camp, and S., Pezeshk, 2019, “On the application of machine learning techniques to derive seismic fragility curves”. *Computers and Structures*, Vol. 218, pp. 108–122.
- Mander, J. B., M. J. N. Priestley, and R. Park, 1988, “Theoretical Stress-Strain Model for Confined Concrete”, *Journal of Structural Engineering*, Vol. 114, No. 8, pp. 1804–1826.
- Mangalathu, S., and H. V., Burton, 2019, “Deep learning-based classification of earthquake-impacted buildings using textual damage descriptions”, *International Journal of Disaster Risk Reduction*, Vol. 36, p. 101111.

- Mangalathu, S., H. Sun, C. C. Nweke, Z. Yi, and H. V. Burton, 2020, “Classifying earthquake damage to buildings using machine learning”, *Earthquake Spectra*, Vol. 36, No. 1, pp. 183–208.
- McKenna, F., M. H. Scott, and G. L. Fenves, 2010, “Nonlinear finite-element analysis software architecture using object composition”, *Journal of Computing in Civil Engineering*, Vol. 24, No. 1, pp. 95–107.
- Murru, M., A. Akinici, G. Falcone, S. Pucci, R. Console, and T. Parsons, 2016, “M 7 earthquake rupture forecast and time-dependent probability for the sea of Marmara region, Turkiye”, *Journal of Geophysical Research: Solid Earth*, Vol. 121, pp. 2679–2707.
- Parsons, T., S. Toda, R. S. Stein, A. Barka, and J. H. Dieterich, 2000, “Heightened odds of large earthquakes near Istanbul: An interaction-based probability calculation”, *Science*, Vol. 288, No. 5466, pp. 661–665.
- Parsons, T., 2004, “Recalculated probability of $M \geq 7$ earthquakes beneath the Sea of Marmara, Turkey”. *Journal of Geophysical Research*, Vol. 109, No. 5.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, 2011, “Scikit-learn: Machine learning in Python”, *the Journal of machine Learning research*, Vol. 12, pp. 2825–2830.
- Python Software Foundation. (2020). “Python Language Reference”, version 3.9.
- Quinlan, J.R., 1986, “Induction of decision trees”. *Machine Learning*, Vol. 1, pp. 81–106.

- Roeslin, S., Q. Ma, H. Juárez-Garcia, A. Gómez-Bernal, J. Wicker, and L. Wother-spoon, 2020, “A machine learning damage prediction model for the 2017 Puebla-Morelos, Mexico, earthquake”, *Earthquake Spectra*, Vol. 36, pp. 314–339.
- Salmi, Z.J., M.I. Khodakarami, and F. Behnamfar, 2023, “Development of seismic fragility curves for RC/MR frames using machine learning methods”. *Asian Journal of Civil Engineering*. Vol. 24, No. 3, pp. 823–836.
- Shinozuka, M., M. Q. Feng, J. Lee, and T. Naganuma, 2000, “Statistical Analysis of Fragility Curves”, *Journal of Engineering Mechanics*, Vol. 126, No. 12, pp. 1224–1231.
- Spacone, E., F. C. Filippou and F. F. Taucer, 1996, “Fibre beam-column model for non-linear analysis of RC frames: Part I. formulation”, *Earthquake Engineering and Structural Dynamics*, Vol. 25, No. 7, pp. 711–725.
- Stojadinović, Z., M. Kovačević, D. Marinković, and B. Stojadinović, 2021, “Rapid earthquake loss assessment based on machine learning and representative sampling”, *Earthquake Spectra*, Vol. 38, No. 1, pp. 152–177.
- The MathWorks Inc., 2022, MATLAB version: 9.13.0 (R2022b), Natick, Massachusetts.
- Ulku O., A.T. Atici, F. Yesilirmak, and U. Hancilar, 2022, “Classifying and predicting earthquake damage by using machine learning after the 2020 Elazığ, Turkey, earthquake”, *3rd European Conference on Earthquake Engineering and Seismology*, Bucharest, Romania.
- Vamvatsikos, D., and C.A. Cornell, 2002, “Incremental dynamic analysis”, *Earthquake Engineering and Structural Dynamics*, Vol. 31, No. 3, pp. 491–514.

- Wald, D. J., and T. I. Allen (2007), “Topographic slope as a proxy for seismic site conditions and amplification”, *Bulletin of the Seismological Society of America*, Vol. 97, No. 5, pp. 1379–1395
- Wu, J., and L. Sarno, 2023, “A machine-learning method for deriving state-dependent fragility curves of existing steel moment frames with masonry infills”, *Engineering Structures*, Vol. 276, p. 115345.
- Yeo, I.K., and R.A. Johnson, 2000, “A new family of power transformations to improve normality or symmetry.” *Biometrika*, Vol. 87, No. 4, pp. 954-959.
- Zhang Y., H.V. Burton, H. Sun, and M., Shokrabadi, 2018, “A machine learning framework for assessing post-earthquake structural safety”. *Structural Safety*, Vol. 72, pp. 1–16.
- Zhao, J., F. Jiang, P. Shi, H. Xing, H. Huang, R. Hou, Y. Zhang, P. Yu, X. Lan, D. Rhoades, P. Somerville, K. Irikura, and Y. Fukushima, 2016, “Ground-Motion Prediction Equations for Subduction Slab Earthquakes in Japan Using Site Class and Simple Geometric Attenuation Functions”. *Bulletin of the Seismological Society of America*, Vol. 106, No. 4, pp. 1535–1551.

**APPENDIX A: SCRIPTS OF GENERATING MODEL,
NONLINEAR TIME HISTORY ANALYSIS, DATA
PREPARATION, MACHINE LEARNING ALGORITHM,
AND DERIVING FRAGILITY CURVES**

The source code has been made publicly available through the GitHub repository, accessible via the provided link: <https://github.com/onurulku23/OpenSees>.

APPENDIX B: THE SELECTED GROUND MOTIONS

Table B.1. Example of selected ground motions with scale factors and time domain parameters for period class 1 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
1	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	2703	0.51	0.08	8.4	8.2
1	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.3	0.24	40.0	39.8
1	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3143	0.6	0.23	74.9	75.4
1	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3142	0.5	0.32	43.0	31.2
1	1130	Ekinozu - Kahramanmaras (02.06.2023)	7.7	North	129	2.16	0.33	40.4	55.4
1	1130	S. San Andreas; NM+SM+NSB	7.3	1st	s650	3.75	1.02	135.9	160.9
1	1130	Palos Verdes Connected	8	1st	s313	2.33	0.35	32.2	133.1
1	1130	S. San Andreas; BB+NM+SM+NSB+SSB+BG	8	2nd	s123	0.8	0.39	40.1	112.2
1	1130	Elsinore; W	6.5	1st	PLS	1.72	0.80	42.2	34.6
1	1130	San Jacinto (SB to C)	7.3	2nd	s564	1.84	0.59	47.6	81.7
1	1130	S. San Andreas; PK+CH+CC+BB+NM+SM+NSB+SSB	8	2nd	Chino	0.71	0.32	43.8	48.6

Table B.2. Example of selected ground motions with scale factors and time domain parameters for period class 1 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
1	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	4408	3.69	0.51	123.6	133.0
1	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3144	0.52	0.32	68.4	58.0
1	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	4404	2.66	0.37	37.0	76.2
1	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	2703	0.92	0.15	15.2	14.8
1	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3142	0.9	0.58	77.4	56.2
1	560	Simi-Santa Rosa	6.5	1st	s066	1.9	0.54	104.7	53.1
1	560	Palos Verdes Connected	8	2nd	s313	2.48	0.57	61.1	61.4
1	560	Cleghorn	6.5	2nd	s648	4.56	1.42	107.1	72.2
1	560	Sierra Madre Connected	6.5	2nd	s482	2.94	1.17	52.9	30.4
1	560	S. San Andreas	8	1st	s603	3.7	0.63	139.4	172.5
1	560	Puente Hills (LA)	7.3	1st	s351	0.84	0.62	79.7	37.1

Table B.3. Example of selected ground motions with scale factors and time domain parameters for period class 1 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
1	270	Izmit Earthquake (17.08.1999)	7.4	North	8101	1.18	0.38	63.3	48.1
1	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3123	0.91	0.60	170.0	57.9
1	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.65	0.51	86.7	86.2
1	270	Duzce Earthquake (12.11.1999)	7.1	East	1401	1.3	1.07	85.6	14.6
1	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3145	1.01	0.61	113.5	115.3
1	270	San Jacinto;SBV+SJV+A+CC+B	8	1st	SGR1T	4.62	0.74	174.9	206.8
1	270	Chino, alt 1	6.5	1st	Chino	1.84	0.84	239.9	120.2
1	270	Puente Hills (LA)	7.3	1st	s351	1.01	0.75	95.9	44.6
1	270	S. San Andreas;SM+NSB	7.3	1st	s285	2.83	2.01	162.6	212.0
1	270	Simi-Santa Rosa	6.5	1st	s066	2.29	0.66	126.2	64.0
1	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB+SSB+B	8	2nd	s351	4.99	0.65	79.0	265.8

Table B.4. Example of selected ground motions with scale factors and time domain parameters for period class 2 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
2	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3135	0.31	0.42	20.3	15.0
2	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3124	0.33	0.21	32.0	29.5
2	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	4404	2.24	0.31	31.2	64.1
2	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3144	0.35	0.22	46.0	39.0
2	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3125	0.57	0.45	42.5	37.6
2	1130	Elsinore; W+GI+T+J	7.3	2nd	s682	2.48	0.66	37.4	38.1
2	1130	S. San Andreas; SM+NSB	7.3	1st	s167	3.98	1.03	133.5	177.6
2	1130	S. San Andreas	8	1st	s603	1.76	0.30	66.3	82.1
2	1130	Santa Monica Connected alt 2	7.3	1st	s313	0.59	0.38	27.0	24.7
2	1130	Chino, alt 2	6.5	1st	Pleasants Peak	2.56	0.73	52.6	60.4
2	1130	S. San Andreas	8	2nd	s313	1.32	0.29	31.3	53.9

Table B.5. Example of selected ground motions with scale factors and time domain parameters for period class 2 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
2	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3144	0.67	0.41	88.1	74.7
2	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3139	0.75	0.39	108.9	89.2
2	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3144	0.48	0.38	64.0	63.7
2	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3135	0.58	0.78	38.0	28.1
2	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	2703	1.63	0.26	26.9	26.2
2	560	S. San Andreas	6.5	1st	s167	4.76	1.54	225.7	119.0
2	560	Cleghorn	6.5	2nd	s648	3.84	1.19	90.2	60.8
2	560	S. San Andreas; NSB+SSB+BG+CO	8	2nd	Seven Oaks Dam	0.98	0.49	103.0	182.6
2	560	S. San Andreas PK+CH+CC+BB+NM+SM+NSB+SSB	8	2nd	s351	1.77	0.47	37.2	64.7
2	560	Puente Hills (LA)	7.3	2nd	s351	1.26	0.84	82.6	28.4
2	560	S. San Andreas	8	1st	s603	3.33	0.56	125.5	155.3

Table B.6. Example of selected ground motions with scale factors and time domain parameters for period class 2 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
2	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3123	0.72	0.48	134.5	45.8
2	270	Izmit Earthquake (17.08.1999)	7.4	North	8101	1.66	0.53	89.1	67.6
2	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3135	0.76	1.02	49.7	36.8
2	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3124	0.81	0.51	78.6	72.4
2	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.63	0.50	84.0	83.6
2	270	S. San Andreas;NSB+SSB+BG+CO	8	2nd	Seven Oaks Dam	1.28	0.64	134.5	238.5
2	270	S. San Andreas;BB+NM+SM+NSB	8	1st	s123	2.17	2.23	459.9	573.7
2	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB+SSB	8	2nd	Chino	1.42	0.64	87.5	97.3
2	270	S. San Andreas	8	1st	Seven Oaks Dam	1.55	0.82	121.3	158.3
2	270	Simi-Santa Rosa	6.5	1st	s066	2.3	0.66	126.8	64.2
2	270	S. San Andreas	8	1st	s603	4.33	0.73	163.1	201.9

Table B.7. Example of selected ground motions with scale factors and time domain parameters for period class 3 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
3	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3135	0.27	0.36	17.7	13.1
3	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3125	0.6	0.47	44.8	39.6
3	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.34	0.27	45.3	45.1
3	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3131	0.72	0.26	34.6	37.6
3	1130	Izmit Earthquake (17.08.1999)	7.4	East	4106	1.13	0.16	40.4	47.5
3	1130	S. San Andreas	8	1st	s399	2.36	0.63	63.2	150.1
3	1130	S. San Andreas	7.3	2nd	s380	2.79	1.10	114.8	269.1
3	1130	S. San Andreas;CH+CC+BB	8	1st	s167	2.68	0.25	67.1	73.5
3	1130	Malibu Coast, alt 2	6.5	1st	s195	1.05	0.35	30.5	22.0
3	1130	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB+SSB	8	1st	s351	1.22	0.41	31.4	38.5
3	1130	San Gabriel	7.3	1st	s313	1.66	0.50	50.0	74.1

Table B.8. Example of selected ground motions with scale factors and time domain parameters for period class 3 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
3	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3139	0.36	0.18	52.3	42.8
3	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3132	1.4	0.71	94.4	44.5
3	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.66	0.52	88.0	87.6
3	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3135	0.52	0.70	34.0	25.2
3	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3144	0.64	0.39	84.1	71.4
3	560	Sierra Madre	6.5	2nd	Rio Hondo	1.83	0.44	56.3	25.4
3	560	S. San Andreas PK+CH+CC+BB+NM+SM+NSB+SSB+BG	8	2nd	s443	4.49	1.50	240.9	769.1
3	560	S. San Andreas;CC+BB+NM+SM+NSB+SSB+BG	8	1st	Rio Hondo	1.96	0.55	86.0	77.6
3	560	Raymond	6.5	1st	s351	1.96	0.59	43.7	32.1
3	560	S. San Andreas	8	1st	Seven Oaks Dam	1.18	0.63	92.3	120.5
3	560	Garlock;GC+GW	8	1st	Antelope	4.39	0.45	78.0	216.3

Table B.9. Example of selected ground motions with scale factors and time domain parameters for period class 3 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
3	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3140	1.22	0.27	96.5	102.5
3	270	Izmit Earthquake (17.08.1999)	7.4	East	8101	2.25	0.84	126.8	52.1
3	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.89	0.70	118.7	118.1
3	270	Izmit Earthquake (17.08.1999)	7.4	North	8101	1.74	0.56	93.4	70.9
3	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3142	0.73	0.47	62.8	45.6
3	270	S. San Andreas;NSB+SSB+BG+CO	8	2nd	Seven Oaks Dam	1.44	0.71	151.3	268.3
3	270	S. San Andreas;NM+SM+NSB+SSB+BG	8	1st	s248	2.52	2.13	378.8	838.0
3	270	S. San Andreas;CC+BB+NM+SM	8	2nd	s526	2.13	0.92	472.6	750.2
3	270	Sierra Madre	6.5	2nd	Rio Hondo	2.46	0.60	75.7	34.2
3	270	S. San Andreas	8	1st	Seven Oaks Dam	1.58	0.84	123.6	161.4
3	270	S. San Andreas;BB+NM+SM+NSB+SSB	8	1st	s566	1.04	0.80	196.8	154.7

Table B.10. Example of selected ground motions with scale factors and time domain parameters for period class 4 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
4	1130	Ekinozu - Kahramanmaraş (02.06.2023)	7.7	East	4617	1.87	0.16	51.2	69.0
4	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3142	0.34	0.22	29.2	21.2
4	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3131	0.76	0.28	36.5	39.7
4	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3145	0.52	0.31	58.4	59.3
4	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3125	0.24	0.26	24.6	22.8
4	1130	Elsinore;T+J+CM	7.3	2nd	s764	1.59	0.71	45.4	128.5
4	1130	S. San Andreas;NM+SM+NSB+SSB+BG	8	2nd	s566	0.81	0.66	125.8	263.2
4	1130	S. San Andreas;CH+CC+BB	8	1st	s167	1.98	0.18	49.5	54.3
4	1130	S. San Andreas;NM+SM+NSB	7.3	2nd	s484	2.12	0.44	33.9	82.2
4	1130	S. San Andreas	7.3	1st	s605	1.7	0.54	43.9	91.0
4	1130	S. San Andreas;CC+BB+NM+SM+NSB+SSB+BG	8	1st	Rio Hondo	1.17	0.33	51.4	46.3

Table B.11. Example of selected ground motions with scale factors and time domain parameters for period class 4 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
4	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3135	0.49	0.66	32.1	23.7
4	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3144	0.69	0.42	90.7	76.9
4	560	Ekinozu - Kahramanmaras (02.06.2023)	7.7	North	4617	3.11	0.18	70.3	67.3
4	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3125	0.47	0.51	48.2	44.6
4	560	Izmit Earthquake (17.08.1999)	7.4	East	8101	1.59	0.59	89.6	36.8
4	560	S. San Andreas;NM+SM+NSB	7.3	1st	s484	4.31	0.98	61.9	68.6
4	560	San Jacinto (SB to C)	7.3	2nd	Moreno valley	1.83	1.08	225.0	183.7
4	560	S. San Andreas;CH+CC+BB	8	1st	s167	3.9	0.36	97.6	107.0
4	560	S. San Andreas;BB+NM+SM+NSB	8	1st	s313	3.09	0.65	79.9	177.1
4	560	Anacapa-Dume, alt 2	7.3	1st	p24	0.8	0.49	70.0	70.1
4	560	Garlock;GC+GW	8	1st	Antelope	4.14	0.42	73.6	204.0

Table B.12. Example of selected ground motions with scale factors and time domain parameters for period class 4 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
4	270	Ekinozu - Kahramanmaras (02.06.2023)	7.7	North	4617	4.3	0.24	97.3	93.1
4	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3132	1.41	0.72	95.1	44.8
4	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3139	0.74	0.38	107.5	88.0
4	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.92	0.73	122.7	122.1
4	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3144	0.95	0.58	124.9	105.9
4	270	S. San Andreas; NSB+SSB+BG+CO	8	1st	Seven Oaks Dam	0.7	0.50	68.2	59.1
4	270	S. San Andreas	7.3	2nd	s692	4.75	1.70	286.0	411.7
4	270	S. San Andreas; CC+BB+NM+SM	8	2nd	s526	1.94	0.84	430.4	683.3
4	270	Anacapa-Dume, alt 2	7.3	1st	p24	1.11	0.68	97.2	97.3
4	270	Chino, alt 1	6.5	1st	Chino	1.45	0.67	189.0	94.8
4	270	Simi-Santa Rosa	6.5	1st	s066	2.1	0.73	95.2	62.2

Table B.13. Example of selected ground motions with scale factors and time domain parameters for period class 5 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
5	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3125	0.29	0.32	29.8	27.5
5	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3135	0.47	0.63	30.8	22.7
5	1130	Ekinozu - Kahramanmaraş (02.06.2023)	7.7	East	4617	2.63	0.22	71.9	97.1
5	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3144	0.51	0.31	67.0	56.9
5	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3145	0.25	0.18	38.4	26.8
5	1130	Elsinore	7.3	1st	s766	1.27	0.74	80.7	56.6
5	1130	S. San Andreas	7.3	1st	Silverwood Lake	2.43	0.90	140.8	119.0
5	1130	S. San Andreas;CH+CC+BB	8	1st	s167	2.02	0.19	50.5	55.4
5	1130	S. San Andreas;SM	7.3	2nd	s165	1.18	0.36	30.5	50.9
5	1130	S. San Andreas PK+CH+CC+BB+NM+SM+NSB+SSB+BG	8	2nd	s351	2.65	0.34	42.0	141.1
5	1130	San Jacinto;SBV+SJV+A+CC+B+SM	8	1st	s765	2.33	0.33	64.6	208.5

Table B.14. Example of selected ground motions with scale factors and time domain parameters for period class 5 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
5	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3145	0.49	0.35	75.3	52.6
5	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3125	0.6	0.47	44.8	39.6
5	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3144	0.66	0.52	88.0	87.6
5	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3139	0.61	0.31	88.6	72.6
5	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3131	0.67	0.24	32.2	35.0
5	560	Puente Hills (LA)	7.3	1st	s351	0.69	0.51	65.5	30.5
5	560	San Jacinto;SJV+A+C	7.3	2nd	MRVY	3.49	2.07	218.3	272.3
5	560	S. San Andreas;PK+CH+CC+BB+NM+SM	8	2nd	Pedley	1.8	0.44	103.5	133.8
5	560	S. San Andreas	8	1st	Seven Oaks Dam	1.14	0.61	89.2	116.4
5	560	Clamshell-Sawpit	6.5	1st	Rio Hondo	1.55	0.63	98.0	49.9
5	560	S. San Andreas;BB+NM+SM+NSB+SSB	8	1st	s566	0.77	0.59	145.7	114.5

Table B.15. Example of selected ground motions with scale factors and time domain parameters for period class 5 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
5	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3145	0.7	0.49	107.6	75.2
5	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3144	1.46	0.90	191.9	162.8
5	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3140	2.01	0.44	158.9	168.8
5	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3139	0.88	0.45	127.8	104.7
5	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3131	0.96	0.35	43.1	25.2
5	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s526	1.87	0.80	258.9	367.0
5	270	S. San Andreas	8	1st	s228	2.68	1.84	156.6	201.9
5	270	S. San Andreas;NSB+SSB+BG+CO	8	1st	Seven Oaks Dam	0.75	0.54	73.1	63.3
5	270	S. San Andreas	8	2nd	Chino	2.16	0.78	150.1	150.0
5	270	Chino, alt 1	6.5	1st	Chino	1.51	0.69	196.9	98.7
5	270	S. San Andreas;NSB+SSB+BG+CO	8	2nd	Seven Oaks Dam	1.48	0.73	155.5	275.8

Table B.16. Example of selected ground motions with scale factors and time domain parameters for period class 6 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
6	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3125	0.26	0.28	26.7	24.7
6	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3135	0.52	0.70	34.0	25.2
6	1130	Ekinozu - Kahramanmaraş (02.06.2023)	7.7	East	4617	2.53	0.21	69.2	93.4
6	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3144	0.37	0.29	49.3	49.1
6	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3131	0.44	0.16	19.8	11.5
6	1130	S. San Andreas; PK+CH+CC+BB+NM+SM+NSB	8	1st	s526	0.61	0.26	84.5	119.7
6	1130	S. San Andreas; NM+SM+NSB+SSB+BG	8	2nd	s566	0.71	0.56	75.9	129.7
6	1130	San Jacinto (SB to C)	6.5	2nd	UCR	2.71	0.68	45.2	34.8
6	1130	S. San Andreas	8	2nd	s545	1.78	1.03	79.2	201.0
6	1130	Clamshell-Sawpit	6.5	2nd	RIO	1.01	0.25	42.7	23.1
6	1130	S. San Andreas; BB+NM+SM+NSB+SSB+BG+CO	8	1st	s732	0.55	0.28	65.3	45.9

Table B.17. Example of selected ground motions with scale factors and time domain parameters for period class 6 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
6	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3124	0.32	0.20	31.0	28.6
6	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3115	0.95	0.28	39.1	28.8
6	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3144	0.75	0.59	100.0	99.5
6	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3145	0.96	0.58	107.9	109.6
6	560	Izmit Earthquake (17.08.1999)	7.4	East	8101	1.26	0.47	71.0	29.2
6	560	North Frontal (West)	7.3	2nd	s690	2.21	1.14	132.9	97.9
6	560	S. San Andreas;CH+CC+BB	8	1st	s167	4.13	0.38	103.3	113.3
6	560	Puente Hills (LA)	7.3	1st	s351	0.59	0.44	56.0	26.1
6	560	S. San Andreas;CC+BB+NM+SM+NSB+SSB	8	2nd	s668	1.82	1.53	107.6	299.7
6	560	Sierra Madre	7.3	1st	s560	1.11	0.49	87.0	62.4
6	560	Simi-Santa Rosa	6.5	1st	s066	1.48	0.52	67.1	43.9

Table B.18. Example of selected ground motions with scale factors and time domain parameters for period class 6 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
6	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3140	2.33	0.51	184.3	195.7
6	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3131	2.07	0.75	99.4	108.1
6	270	Golyaka - Duzce (23.11.2023)	5.9	East	8102	1.65	0.68	121.5	38.6
6	270	Izmit Earthquake (17.08.1999)	7.4	North	8101	2.7	0.87	144.9	109.9
6	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3124	0.49	0.31	47.5	43.8
6	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s526	1.87	0.80	258.9	367.0
6	270	Sierra Madre	7.3	1st	s560	1.7	0.75	133.3	95.6
6	270	S. San Andreas;NSB+SSB+BG+CO	8	1st	Seven Oaks Dam	1.02	0.73	99.4	86.1
6	270	S. San Andreas;CC+BB+NM+SM+NSB+SSB	8	2nd	s668	2.77	2.33	163.8	456.2
6	270	S. San Andreas;CC+BB+NM+SM	8	1st	s526	1.27	1.02	264.8	137.4
6	270	S. San Andreas	7.3	1st	s167	1.99	0.81	155.8	136.4

Table B.19. Example of selected ground motions with scale factors and time domain parameters for period class 7 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
7	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	2703	1.09	0.18	18.0	17.5
7	1130	Ekinozu - Kahramanmaraş (02.06.2023)	7.7	East	4617	2.12	0.18	58.0	78.2
7	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	4616	0.48	0.24	40.4	34.2
7	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3131	0.67	0.24	30.1	17.6
7	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3144	0.35	0.28	46.7	46.4
7	1130	S. San Andreas;CH+CC+BB	8	1st	s167	1.22	0.22	151.6	131.6
7	1130	Chino, alt 1	6.5	2nd	s636	2.17	0.51	49.1	24.7
7	1130	San Jacinto;SJV+A+C	8	1st	s720	2.87	0.30	67.5	60.8
7	1130	San Jacinto;SBV+SJV+A+CC+B+SM	8	2nd	s765	2.01	0.30	54.1	203.6
7	1130	S. San Andreas	6.5	1st	s465	1.95	0.78	56.0	31.6
7	1130	S. San Andreas	7.3	2nd	s528	2.63	0.51	58.3	146.6

Table B.20. Example of selected ground motions with scale factors and time domain parameters for period class 7 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
7	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3135	0.82	1.10	53.7	39.7
7	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3124	0.28	0.18	27.2	25.0
7	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.69	0.54	92.0	91.5
7	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3131	1.33	0.48	59.7	34.9
7	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3139	0.61	0.31	88.6	72.6
7	560	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s526	1.06	0.45	146.8	208.0
7	560	S. San Andreas	6.5	1st	s778	4.79	1.38	90.5	77.5
7	560	Sierra Madre	7.3	1st	s560	1.24	0.55	97.2	69.7
7	560	S. San Andreas	8	1st	Seven Oaks Dam	0.82	0.44	64.2	83.7
7	560	S. San Andreas;NSB+SSB+BG+CO	8	1st	Seven Oaks Dam	0.7	0.50	68.2	59.1
7	560	San Gabriel	7.3	1st	s313	4.36	1.31	131.2	194.7

Table B.21. Example of selected ground motions with scale factors and time domain parameters for period class 7 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
7	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3131	2.08	0.75	93.4	54.5
7	270	Izmit Earthquake (17.08.1999)	7.4	North	8101	2.42	0.78	129.9	98.5
7	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3144	1.09	0.86	145.4	144.6
7	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3124	0.43	0.27	41.7	38.4
7	270	Golyaka - Duzce (23.11.2023)	5.9	East	8102	1.45	0.59	106.7	33.9
7	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s526	1.66	0.71	229.9	325.8
7	270	Santa Monica Connected alt 2	6.5	2nd	p25	3.78	2.54	222.1	195.0
7	270	S. San Andreas;CC+BB+NM+SM	8	1st	s526	1.24	1.00	258.6	134.2
7	270	S. San Andreas	8	1st	Seven Oaks Dam	1.28	0.68	100.2	130.7
7	270	Chino, alt 1	6.5	2nd	Chino	1.91	0.80	147.7	59.8
7	270	S. San Andreas	7.3	1st	s167	1.99	0.81	155.8	136.4

Table B.22. Example of selected ground motions with scale factors and time domain parameters for period class 8 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
8	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3142	0.4	0.26	34.4	25.0
8	1130	Ekinozu - Kahramanmaras (02.06.2023)	7.7	East	4617	1.73	0.15	47.3	63.8
8	1130	Izmit Earthquake (17.08.1999)	7.4	East	4106	2.44	0.35	87.2	102.6
8	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3144	0.33	0.20	43.4	36.8
8	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3143	0.33	0.12	34.4	29.9
8	1130	S. San Andreas;CH+CC+BB+NM+SM+NSB	8	1st	s566	0.32	0.23	59.7	95.1
8	1130	S. San Andreas;SM+NSB+SSB	7.3	1st	s246	1.44	0.58	90.6	92.8
8	1130	S. San Andreas;CC+BB+NM+SM+NSB+SSB+BG	8	1st	Rio Hondo	1.07	0.30	47.0	42.3
8	1130	San Gabriel	6.5	1st	s161	0.88	0.27	40.4	10.8
8	1130	S. San Andreas	8	1st	s401	1.55	0.40	112.2	239.6
8	1130	S. San Andreas	8	2nd	s545	1.7	0.98	75.6	191.9

Table B.23. Example of selected ground motions with scale factors and time domain parameters for period class 8 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
8	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3145	1.21	0.73	136.0	138.1
8	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3139	0.36	0.18	52.3	42.8
8	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3131	0.77	0.28	34.6	20.2
8	560	Izmit Earthquake (17.08.1999)	7.4	East	8101	1.11	0.41	62.6	25.7
8	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3144	0.65	0.40	85.4	72.5
8	560	S. San Andreas;CH+CC+BB+NM+SM+NSB	8	1st	s566	0.63	0.45	117.6	187.1
8	560	S. San Andreas;SM+NSB+SSB	7.3	2nd	s246	2.62	1.24	133.6	175.8
8	560	Chino, alt 1	6.5	1st	Chino	0.87	0.40	113.4	56.9
8	560	S. San Andreas	8	1st	Seven Oaks Dam	0.82	0.44	64.2	83.7
8	560	S. San Andreas;CH+CC+BB	8	1st	s167	4.14	0.38	103.6	113.6
8	560	S. San Andreas	8	1st	s313	3.5	0.96	70.9	120.9

Table B.24. Example of selected ground motions with scale factors and time domain parameters for period class 8 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
8	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3144	1.07	0.66	140.6	119.3
8	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3140	0.97	0.21	76.7	81.5
8	270	Izmit Earthquake (17.08.1999)	7.4	North	8101	2.38	0.76	127.7	96.9
8	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	3124	0.85	0.49	95.5	40.2
8	270	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3139	0.59	0.30	85.7	70.2
8	270	S. San Andreas;CH+CC+BB+NM+SM+NSB	8	1st	s566	1.04	0.74	194.1	308.9
8	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s566	1.09	0.73	329.1	411.7
8	270	Chino, alt 1	6.5	2nd	Chino	1.71	0.71	132.3	53.6
8	270	Puente Hills	7.3	2nd	s351	4.59	1.41	133.4	97.9
8	270	S. San Andreas	7.3	1st	s167	1.98	0.80	155.0	135.7
8	270	Chino, alt 1	6.5	1st	Chino	1.43	0.66	186.4	93.4

Table B.25. Example of selected ground motions with scale factors and time domain parameters for period class 9 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
9	1130	Izmit Earthquake (17.08.1999)	7.4	East	4106	1.71	0.25	61.1	71.9
9	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3125	0.32	0.35	32.8	30.4
9	1130	Ekinozu - Kahramanmaraş (02.06.2023)	7.7	East	4617	1.44	0.12	39.4	53.1
9	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	4404	1.88	0.26	37.4	76.5
9	1130	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3131	0.46	0.17	20.7	12.1
9	1130	S. San Andreas	6.5	1st	ALIS	4.5	0.55	52.8	53.9
9	1130	S. San Andreas	7.3	2nd	s380	2.35	0.93	96.7	226.6
9	1130	S. San Andreas;CH+CC+BB+NM+SM+NSB	8	1st	s648	0.26	0.24	102.7	87.4
9	1130	Sierra Madre	7.3	1st	s560	0.57	0.25	44.7	32.1
9	1130	Anacapa-Dume, alt 2	7.3	2nd	p24	0.85	0.49	75.6	64.3
9	1130	S. San Andreas;CC+BB+NM+SM+NSB+SSB+BG	8	1st	Rio Hondo	1.03	0.29	45.2	40.8

Table B.26. Example of selected ground motions with scale factors and time domain parameters for period class 9 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
9	560	Izmit Earthquake (17.08.1999)	7.4	East	8101	1.16	0.43	65.4	26.9
9	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3125	0.63	0.69	64.7	59.8
9	560	Ekinozu - Kahramanmaraş (02.06.2023)	7.7	East	4617	2.84	0.24	77.7	104.8
9	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	North	4404	3.71	0.51	73.9	151.0
9	560	Pazarcik-Kahramanmaraş (02.06.2023)	7.8	East	3131	0.9	0.33	40.4	23.6
9	560	San Jose	6.5	2nd	s520	1.32	1.21	115.1	26.9
9	560	S. San Andreas	7.3	2nd	s692	3.17	1.14	190.9	274.8
9	560	S. San Andreas;BB+NM+SM+NSB+SSB+BG+CO	8	2nd	RIO	1.5	0.35	142.7	212.5
9	560	S. San Andreas	7.3	1st	s167	1.22	0.50	95.5	83.6
9	560	S. San Andreas;CC+BB+NM+SM+NSB+SSB+BG	8	1st	Rio Hondo	2.03	0.57	89.1	80.3
9	560	S. San Andreas;CC+BB+NM+SM	8	1st	s526	0.59	0.47	123.0	63.8

Table B.27. Example of selected ground motions with scale factors and time domain parameters for period class 9 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
9	270	Ekinozu - Kahramanmaras (02.06.2023)	7.7	East	4617	4.8	0.40	131.3	177.2
9	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3143	1.89	0.67	197.2	171.2
9	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3140	1.19	0.26	94.1	100.0
9	270	Izmit Earthquake (17.08.1999)	7.4	East	8101	1.97	0.73	111.1	45.6
9	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3145	0.57	0.40	87.6	61.2
9	270	S. San Andreas	7.3	1st	s692	4.81	1.81	152.7	182.1
9	270	S. San Andreas;CC+BB+NM+SM	8	2nd	s526	1.85	0.80	410.4	651.6
9	270	S. San Andreas;BB+NM+SM+NSB+SSB+BG+CO	8	2nd	RIO	2.53	0.59	240.6	358.5
9	270	Sierra Madre	7.3	2nd	s560	1.42	0.66	143.5	64.3
9	270	S. San Andreas;CC+BB+NM+SM	8	1st	s526	0.99	0.80	206.4	107.1
9	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s566	1	0.67	301.9	377.7

Table B.28. Example of selected ground motions with scale factors and time domain parameters for period class 10 and Vs30 1130 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
10	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	201	1.47	0.41	54.0	30.8
10	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3140	0.58	0.13	45.9	48.7
10	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	4404	2.96	0.40	58.9	120.5
10	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3131	0.59	0.21	26.5	15.5
10	1130	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.35	0.28	46.7	46.4
10	1130	S. San Andreas	7.3	2nd	s380	3.29	0.93	111.3	149.5
10	1130	S. San Andreas	8	1st	s778	2.36	0.25	63.5	88.5
10	1130	S. San Andreas	8	1st	s380	2.31	0.40	118.4	148.0
10	1130	S. San Andreas;BB+NM+SM+NSB+SSB	8	2nd	s566	0.75	0.39	59.5	112.9
10	1130	Chino, alt 1	6.5	1st	Chino	0.38	0.17	49.5	24.8
10	1130	S. San Andreas	8	1st	s228	0.79	0.33	58.5	101.5

Table B.29. Example of selected ground motions with scale factors and time domain parameters for period class 10 and Vs30 560 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
10	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	4616	0.96	0.48	80.8	68.4
10	560	Ekinozu - Kahramanmaras (02.06.2023)	7.7	East	4617	2.27	0.19	62.1	83.8
10	560	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	0.69	0.54	92.0	91.5
10	560	Golyaka - Duzce (23.11.2023)	5.9	North	8102	1.43	0.31	50.3	24.9
10	560	Izmit Earthquake (17.08.1999)	7.4	East	8101	1.35	0.50	76.1	31.3
10	560	S. San Andreas	8	2nd	s778	4.59	1.78	233.0	492.6
10	560	S. San Andreas;NSB+SSB+BG+CO	8	1st	Seven Oaks Dam	0.69	0.50	67.2	58.2
10	560	Chino, alt 1	6.5	1st	Chino	0.75	0.34	97.8	49.0
10	560	S. San Andreas	7.3	1st	s228	2.79	0.58	99.6	127.1
10	560	Sierra Madre	7.3	1st	s560	1.01	0.45	79.2	56.8
10	560	S. San Andreas;CC+BB+NM+SM+NSB+SSB+BG	8	2nd	RIO	1.27	0.48	148.3	199.4

Table B-30. Example of selected ground motions with scale factors and time domain parameters for period class 10 and Vs30 270 m/s.

Period Class	Vs30	Earthquake	Mw	Component	Station	SF	PGA (g)	PGV (cm/s)	PGD (cm)
10	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3140	1.96	0.43	155.0	164.6
10	270	Ekinozu - Kahramanmaras (02.06.2023)	7.7	East	4617	3.93	0.33	107.5	145.0
10	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	East	3144	1.2	0.95	160.0	159.2
10	270	Pazarcik-Kahramanmaras (02.06.2023)	7.8	North	3140	1.07	0.21	67.8	51.5
10	270	Izmit Earthquake (17.08.1999)	7.4	East	1612	2.82	0.35	79.7	84.2
10	270	Santa Monica Connected alt 2	6.5	1st	P25	3.87	1.57	142.6	80.6
10	270	S. San Andreas;BB+NM+SM+NSB+SSB+BG+CO	8	2nd	RIO	2.24	0.52	213.0	317.4
10	270	S. San Andreas;PK+CH+CC+BB+NM+SM+NSB	8	1st	s566	0.97	0.65	292.9	366.4
10	270	S. San Andreas;CC+BB+NM+SM	8	1st	s526	0.9	0.72	187.7	97.4
10	270	Sierra Madre	7.3	2nd	s560	1.19	0.55	120.3	53.9
10	270	S. San Andreas;CC+BB+NM+SM	8	2nd	s526	1.69	0.73	374.9	595.2

APPENDIX C: CONFUSION MATRICES FOR ALL ML ALGORITHMS AND ALL INTENSITY MEASURES

Table C.1. Confusion matrix of the DT algorithm with PGA.

DT – PGA	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
0.0%-0.1%	1344	138	4	0	0	0	0	0	0	0	0	0	1486	0.90
0.1%-0.2%	161	2635	193	12	3	3	0	0	0	0	1	0	3008	0.88
0.2%-0.3%	8	225	1835	216	16	4	3	1	0	0	0	0	2308	0.80
0.3%-0.4%	0	8	240	1408	241	10	6	1	1	1	1	4	1920	0.73
0.4%-0.5%	0	2	21	229	988	165	24	7	8	3	9	22	1456	0.68
0.5%-0.6%	0	1	7	28	242	828	176	37	12	5	22	42	1358	0.61
0.6%-0.7%	0	1	3	13	35	218	678	170	23	11	42	42	1194	0.57
0.7%-0.8%	0	1	6	2	15	30	174	498	115	20	42	42	903	0.55
0.8%-0.9%	0	1	5	1	6	17	35	146	385	129	84	84	809	0.48
0.9%-1.0%	0	1	0	2	3	8	14	39	109	299	190	190	665	0.45
>1.0%	0	0	3	8	19	27	47	77	98	182	4803	4803	5264	0.91
Total	1513	3013	2317	1919	1568	1310	1157	976	751	651	5196	5196	0.77	
Precision	0.89	0.87	0.79	0.73	0.63	0.63	0.59	0.51	0.51	0.46	0.92	0.92		

Table C.2. Confusion matrix of the DT algorithm with PGV.

DT – PGV		Predicted Class											Total	Recall	
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	1333	150	2	1	0	0	0	0	0	0	0	1486	0.90
		0.1%-0.2%	162	2620	201	15	4	1	4	1	0	0	0	3008	0.87
		0.2%-0.3%	9	220	1862	198	9	4	2	1	0	2	1	2308	0.81
		0.3%-0.4%	2	13	252	1412	199	25	6	4	0	0	7	1920	0.74
		0.4%-0.5%	0	5	25	229	965	181	32	6	4	0	9	1456	0.66
		0.5%-0.6%	0	3	11	52	233	817	183	29	14	1	15	1358	0.60
		0.6%-0.7%	0	4	2	5	34	221	722	151	25	2	28	1194	0.60
		0.7%-0.8%	0	2	4	5	12	38	167	501	115	20	39	903	0.55
		0.8%-0.9%	0	0	7	4	7	15	35	148	405	110	78	809	0.50
		0.9%-1.0%	0	2	0	1	0	8	12	34	127	283	198	665	0.43
>1.0%		0	0	7	10	14	32	59	66	113	167	4796	5264	0.91	
Total		1506	3019	2373	1932	1477	1342	1222	941	803	585	5171	0.77		
Precision		0.89	0.87	0.78	0.73	0.65	0.61	0.59	0.53	0.50	0.48	0.93			

Table C.3. Confusion matrix of the DT algorithm with PGD.

DT – PGD		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	1354	127	4	0	0	0	0	0	1	0	0	1486	0.91
		0.1%-0.2%	134	2671	186	6	5	0	0	0	1	0	5	3008	0.89
		0.2%-0.3%	2	206	1855	203	16	7	4	6	4	0	5	2308	0.80
		0.3%-0.4%	1	4	265	1390	208	24	12	1	2	0	13	1920	0.72
		0.4%-0.5%	2	1	16	230	975	180	23	11	3	0	15	1456	0.67
		0.5%-0.6%	1	1	10	39	238	819	190	26	9	1	24	1358	0.60
		0.6%-0.7%	1	1	4	4	42	214	669	166	46	4	43	1194	0.56
		0.7%-0.8%	0	1	4	5	16	39	156	508	115	15	44	903	0.56
		0.8%-0.9%	0	0	4	6	12	12	34	165	390	95	91	809	0.48
		0.9%-1.0%	0	1	1	7	3	7	16	42	144	228	216	665	0.34
>1.0%		0	2	8	7	11	21	62	75	138	167	4773	5264	0.91	
Total		1495	3015	2357	1897	1526	1323	1166	1000	853	510	5229	0.77		
Precision		0.91	0.89	0.79	0.73	0.64	0.62	0.57	0.51	0.46	0.45	0.91			

Table C.4. Confusion matrix of the DT algorithm with $S_a(T_1)$.

DT - S _a (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%		
0.0%-0.1%		1023	443	14	5	1	0	0	0	0	0	0	1486	0.69
0.1%-0.2%		306	2132	400	96	42	13	13	4	2	0	0	3008	0.71
0.2%-0.3%		28	628	1056	324	120	63	37	20	6	1	25	2308	0.46
0.3%-0.4%		3	214	428	775	195	126	62	32	15	5	65	1920	0.40
0.4%-0.5%		0	79	224	275	430	172	87	54	15	15	105	1456	0.30
0.5%-0.6%		0	33	113	177	239	352	158	58	24	15	189	1358	0.26
0.6%-0.7%		0	8	79	102	131	158	289	90	44	22	271	1194	0.24
0.7%-0.8%		0	2	25	67	74	78	146	128	46	21	316	903	0.14
0.8%-0.9%		0	2	29	59	56	66	71	81	62	30	353	809	0.08
0.9%-1.0%		0	1	4	39	35	49	46	53	31	43	364	665	0.06
>1.0%		1	4	18	59	101	142	131	179	125	115	4389	5264	0.83
Total		1361	3546	2390	1978	1424	1219	1040	699	370	267	6077	0.52	
Precision		0.75	0.60	0.44	0.39	0.30	0.29	0.28	0.18	0.17	0.16	0.72		

Table C.5. Confusion matrix of the DT algorithm with $S_d(T_1)$.

DT - S _d (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
0.0%-0.1%		1042	423	17	4	0	0	0	0	0	0	0	1486	0.70
0.1%-0.2%		320	2109	436	98	26	7	7	1	1	0	3	3008	0.70
0.2%-0.3%		18	649	1049	335	137	46	34	9	10	2	19	2308	0.45
0.3%-0.4%		1	202	439	716	304	108	51	25	11	7	56	1920	0.37
0.4%-0.5%		0	56	244	310	427	145	85	29	29	16	115	1456	0.29
0.5%-0.6%		0	17	116	200	305	300	136	52	36	18	178	1358	0.22
0.6%-0.7%		0	10	72	112	152	167	268	85	38	26	264	1194	0.22
0.7%-0.8%		0	4	17	73	99	91	127	105	61	31	295	903	0.12
0.8%-0.9%		0	2	19	52	69	55	71	67	63	34	377	809	0.08
0.9%-1.0%		0	0	1	28	45	46	61	26	47	45	366	665	0.07
>1.0%		0	3	13	47	101	118	179	112	114	98	4479	5264	0.85
Total		1381	3475	2423	1975	1665	1083	1019	511	410	277	6152	0.52	
Precision		0.75	0.61	0.43	0.36	0.26	0.28	0.26	0.21	0.15	0.16	0.73		

Table C.6. Confusion matrix of the DT algorithm with CAV.

DT – CAV		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	1361	124	1	0	0	0	0	0	0	0	0	1486	0.92
		0.1%-0.2%	150	2662	182	6	3	1	0	1	0	0	3	3008	0.88
		0.2%-0.3%	2	220	1855	209	9	4	2	0	1	1	5	2308	0.80
		0.3%-0.4%	0	4	267	1409	204	19	12	0	0	2	3	1920	0.73
		0.4%-0.5%	0	4	21	232	971	177	25	7	11	1	7	1456	0.67
		0.5%-0.6%	1	1	12	47	221	844	163	29	11	6	23	1358	0.62
		0.6%-0.7%	1	2	4	4	25	226	690	156	30	9	47	1194	0.58
		0.7%-0.8%	0	0	7	7	15	27	164	501	118	22	42	903	0.55
		0.8%-0.9%	0	0	3	10	5	22	32	150	404	107	76	809	0.50
		0.9%-1.0%	0	1	1	4	4	9	10	35	129	304	168	665	0.46
>1.0%		0	0	12	19	29	33	68	58	110	209	4726	5264	0.90	
Total		1515	3018	2365	1947	1486	1362	1166	937	814	661	5100	0.77		
Precision		0.90	0.88	0.78	0.72	0.65	0.62	0.59	0.53	0.50	0.46	0.93			

Table C.7. Confusion matrix of the RF algorithm with PGV.

RF – PGV		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	1389	96	0	1	0	0	0	0	0	0	0	1486	0.93
		0.1%-0.2%	89	2776	139	3	0	1	0	0	0	0	0	3008	0.92
		0.2%-0.3%	0	125	2037	135	2	2	2	2	0	1	2	2308	0.88
		0.3%-0.4%	1	3	158	1578	167	10	1	0	0	0	2	1920	0.82
		0.4%-0.5%	0	0	5	167	1155	116	5	1	1	0	6	1456	0.79
		0.5%-0.6%	0	1	2	6	154	1012	142	17	1	1	22	1358	0.75
		0.6%-0.7%	0	0	1	4	5	151	812	151	26	3	41	1194	0.68
		0.7%-0.8%	0	1	3	1	0	13	122	602	98	10	53	903	0.67
		0.8%-0.9%	0	0	1	2	1	0	13	122	498	107	65	809	0.62
		0.9%-1.0%	0	0	0	0	1	1	1	3	93	395	163	665	0.59
>1.0%		0	0	2	5	7	21	32	38	55	147	4957	5264	0.94	
Total		1479	3002	2348	1902	1492	1327	1132	942	772	664	5311	0.84		
Precision		0.94	0.92	0.87	0.83	0.77	0.76	0.72	0.64	0.65	0.59	0.93			

Table C.8. Confusion matrix of the RF algorithm with PGD.

RF – PGD		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
0.0%-0.1%		1388	97	0	0	0	1	0	0	0	0	0	1486	0.93	
0.1%-0.2%		92	2775	134	4	1	0	0	0	1	0	1	3008	0.92	
0.2%-0.3%		0	115	2037	141	7	0	3	3	0	0	2	2308	0.88	
0.3%-0.4%		0	3	153	1578	169	11	3	0	0	0	3	1920	0.82	
0.4%-0.5%		0	0	5	183	1138	119	6	1	0	0	4	1456	0.78	
0.5%-0.6%		0	1	1	7	151	1010	149	16	2	1	20	1358	0.74	
0.6%-0.7%		0	0	0	2	10	163	806	145	22	4	42	1194	0.68	
0.7%-0.8%		0	0	3	0	1	13	114	615	93	13	51	903	0.68	
0.8%-0.9%		0	0	1	1	3	0	13	117	501	102	71	809	0.62	
0.9%-1.0%		0	0	2	2	0	2	1	10	92	389	167	665	0.58	
>1.0%		0	0	4	3	8	21	40	46	58	154	4930	5264	0.94	
Total		1480	2991	2340	1921	1488	1340	1135	953	769	663	5291	0.84		
Precision		0.94	0.93	0.87	0.82	0.76	0.75	0.71	0.65	0.65	0.59	0.93			

Table C.9. Confusion matrix of the RF algorithm with $S_a(T_1)$.

RF - S _a (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%		
0.0%-0.1%		1062	393	22	8	1	0	0	0	0	0	0	1486	0.71
0.1%-0.2%		286	2098	455	94	37	20	13	1	0	1	3	3008	0.70
0.2%-0.3%		17	554	1174	303	121	53	44	15	9	1	17	2308	0.51
0.3%-0.4%		2	193	413	799	202	115	65	40	22	10	59	1920	0.42
0.4%-0.5%		0	65	199	260	502	144	98	54	21	23	90	1456	0.34
0.5%-0.6%		0	20	97	168	241	415	131	58	37	16	175	1358	0.31
0.6%-0.7%		0	8	70	79	126	152	320	105	54	41	239	1194	0.27
0.7%-0.8%		0	3	20	59	71	97	116	171	61	31	274	903	0.19
0.8%-0.9%		0	3	20	52	49	57	67	92	95	46	328	809	0.12
0.9%-1.0%		0	0	5	30	29	41	44	49	49	75	343	665	0.11
>1.0%		0	2	15	40	92	100	133	135	101	104	4542	5264	0.86
Total		1367	3339	2490	1892	1471	1194	1031	720	449	348	6070	0.55	
Precision		0.78	0.63	0.47	0.42	0.34	0.35	0.31	0.24	0.21	0.22	0.75		

Table C.10. Confusion matrix of the RF algorithm with $S_d(T_1)$.

RF - S _d (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%			
0.0%-0.1%		1056	407	20	3	0	0	0	0	0	0	0	1486	0.71
0.1%-0.2%		275	2147	427	99	33	12	9	4	0	0	2	3008	0.71
0.2%-0.3%		13	588	1158	293	118	48	46	14	5	2	23	2308	0.50
0.3%-0.4%		0	172	396	825	218	118	70	35	27	6	53	1920	0.43
0.4%-0.5%		1	53	212	293	483	143	63	51	32	19	106	1456	0.33
0.5%-0.6%		0	19	103	175	228	411	124	53	40	25	180	1358	0.30
0.6%-0.7%		0	11	75	82	125	149	276	108	53	38	277	1194	0.23
0.7%-0.8%		0	3	17	62	78	85	126	131	63	30	308	903	0.15
0.8%-0.9%		1	3	14	51	49	60	60	78	85	40	368	809	0.11
0.9%-1.0%		0	0	8	18	40	52	41	36	45	74	351	665	0.11
>1.0%		0	4	19	39	71	118	133	114	103	72	4591	5264	0.87
Total		1346	3407	2449	1940	1443	1196	948	624	453	306	6259	0.55	
Precision		0.78	0.63	0.47	0.43	0.33	0.34	0.29	0.21	0.19	0.24	0.73		

Table C.11. Confusion matrix of the RF algorithm with CAV.

RF – CAV		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
0.0%-0.1%		1388	98	0	0	0	0	0	0	0	0	0	1486	0.93	
0.1%-0.2%		85	2779	137	5	0	1	0	0	0	0	1	3008	0.92	
0.2%-0.3%		1	137	2017	147	3	0	0	2	0	0	1	2308	0.87	
0.3%-0.4%		0	1	150	1595	152	9	8	2	2	0	1	1920	0.83	
0.4%-0.5%		0	0	8	162	1142	121	10	1	1	2	9	1456	0.78	
0.5%-0.6%		0	1	1	3	141	1013	152	16	5	3	23	1358	0.75	
0.6%-0.7%		0	4	0	7	8	158	811	146	21	1	38	1194	0.68	
0.7%-0.8%		0	0	1	0	2	12	117	608	102	13	48	903	0.67	
0.8%-0.9%		0	0	1	2	3	2	12	120	495	106	68	809	0.61	
0.9%-1.0%		0	0	1	0	1	1	2	8	92	388	172	665	0.58	
>1.0%		0	0	2	2	8	25	33	37	53	165	4939	5264	0.94	
Total		1474	3020	2318	1923	1460	1342	1145	940	771	678	5300	0.84		
Precision		0.94	0.92	0.87	0.83	0.78	0.75	0.71	0.65	0.64	0.57	0.93			

Table C.12. Confusion matrix of the KNN algorithm with PGA.

KNN – PGA		Predicted Class											Total	Recall	
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	832	551	86	16	1	0	0	0	0	0	0	1486	0.56
		0.1%-0.2%	408	1664	582	240	74	32	5	1	0	1	1	3008	0.55
		0.2%-0.3%	32	860	830	279	134	68	48	15	6	2	34	2308	0.36
		0.3%-0.4%	5	425	458	448	183	88	64	31	13	11	194	1920	0.23
		0.4%-0.5%	2	200	287	215	252	105	73	34	11	5	272	1456	0.17
		0.5%-0.6%	0	134	189	194	208	177	96	36	20	10	294	1358	0.13
		0.6%-0.7%	0	74	129	126	151	119	128	37	25	8	397	1194	0.11
		0.7%-0.8%	0	24	78	92	100	81	67	43	11	21	386	903	0.05
		0.8%-0.9%	0	8	53	65	54	49	53	24	22	7	474	809	0.03
		0.9%-1.0%	0	8	18	33	34	31	36	23	16	8	458	665	0.01
>1.0%		0	21	46	72	95	97	89	83	53	32	4676	5264	0.89	
Total		1279	3969	2756	1780	1286	847	659	327	177	105	7186	0.45		
Precision		0.65	0.42	0.30	0.25	0.20	0.21	0.19	0.13	0.12	0.08	0.65			

Table C.13. Confusion matrix of the KNN algorithm with PGV.

KNN – PGV		Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%		
Actual Class	0.0%-0.1%	705	650	114	16	1	0	0	0	0	0	0	1486	0.47
	0.1%-0.2%	373	1814	423	244	104	26	13	2	1	0	8	3008	0.60
	0.2%-0.3%	47	821	716	248	147	77	71	48	18	9	106	2308	0.31
	0.3%-0.4%	12	528	379	334	137	81	64	33	22	14	316	1920	0.17
	0.4%-0.5%	0	288	287	158	186	82	51	36	11	6	351	1456	0.13
	0.5%-0.6%	1	159	258	178	141	138	57	27	10	5	384	1358	0.10
	0.6%-0.7%	0	63	252	93	100	100	109	33	19	10	415	1194	0.09
	0.7%-0.8%	0	33	117	89	90	57	67	47	8	9	386	903	0.05
	0.8%-0.9%	0	8	75	84	43	55	47	39	22	10	426	809	0.03
	0.9%-1.0%	0	1	43	53	37	24	40	30	17	3	417	665	0.00
>1.0%	0	15	107	82	75	75	85	59	42	11	4713	5264	0.90	
Total		1138	4380	2771	1579	1061	715	604	354	170	77	7522	0.43	
Precision		0.62	0.41	0.26	0.21	0.18	0.19	0.18	0.13	0.13	0.04	0.63		

Table C.14. Confusion matrix of the KNN algorithm with PGD.

KNN – PGD		Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
0.0%-0.1%		747	618	91	19	6	2	1	0	0	0	2	1486	0.50
0.1%-0.2%		502	1737	367	168	92	49	21	7	6	1	58	3008	0.58
0.2%-0.3%		128	853	650	153	99	51	37	16	8	10	303	2308	0.28
0.3%-0.4%		65	577	316	248	93	71	38	32	10	11	459	1920	0.13
0.4%-0.5%		31	392	229	94	119	52	49	26	13	4	447	1456	0.08
0.5%-0.6%		19	317	187	124	77	95	45	21	14	4	455	1358	0.07
0.6%-0.7%		10	185	223	105	86	49	66	21	13	8	428	1194	0.06
0.7%-0.8%		5	109	154	89	54	36	38	27	10	9	372	903	0.03
0.8%-0.9%		3	64	129	94	41	21	22	19	8	4	404	809	0.01
0.9%-1.0%		3	61	92	47	28	19	20	6	6	8	375	665	0.01
>1.0%		9	145	257	193	110	87	87	51	28	20	4277	5264	0.81
Total		1522	5058	2695	1334	805	532	424	226	116	79	7580	0.39	
Precision		0.49	0.34	0.24	0.19	0.15	0.18	0.16	0.12	0.07	0.10	0.56		

Table C.15. Confusion matrix of the KNN algorithm with $S_a(T_1)$.

KNN – S _a (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
0.0%-0.1%		960	496	26	2	2	0	0	0	0	0	0	1486	0.65
0.1%-0.2%		356	2063	427	97	40	14	8	2	0	0	1	3008	0.69
0.2%-0.3%		23	771	972	320	110	38	41	8	3	1	21	2308	0.42
0.3%-0.4%		2	354	448	675	197	80	48	31	9	3	73	1920	0.35
0.4%-0.5%		0	126	315	287	322	132	84	41	16	8	125	1456	0.22
0.5%-0.6%		0	59	171	238	233	242	129	37	28	10	211	1358	0.18
0.6%-0.7%		0	24	124	102	165	174	181	87	38	18	281	1194	0.15
0.7%-0.8%		0	6	49	86	101	103	115	75	35	23	310	903	0.08
0.8%-0.9%		0	5	37	76	59	76	71	57	30	18	380	809	0.04
0.9%-1.0%		0	1	30	38	46	36	55	42	18	13	386	665	0.02
>1.0%		0	6	48	85	124	149	147	95	56	46	4508	5264	0.86
Total		1341	3911	2647	2006	1399	1044	879	475	233	140	6296	0.49	
Precision		0.72	0.53	0.37	0.34	0.23	0.23	0.21	0.16	0.13	0.09	0.72		

Table C.16. Confusion matrix of the KNN algorithm with $S_d(T_1)$.

KNN – S _d (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%		
0.0%-0.1%		941	512	28	3	2	0	0	0	0	0	0	1486	0.63
0.1%-0.2%		332	2079	434	100	35	15	8	2	0	0	3	3008	0.69
0.2%-0.3%		17	784	983	315	103	32	42	6	1	2	23	2308	0.43
0.3%-0.4%		5	330	443	683	200	99	52	30	10	3	65	1920	0.36
0.4%-0.5%		1	127	312	322	321	118	70	36	16	8	125	1456	0.22
0.5%-0.6%		1	55	168	243	234	248	117	44	26	12	210	1358	0.18
0.6%-0.7%		0	31	124	110	145	178	178	85	35	21	287	1194	0.15
0.7%-0.8%		0	9	51	81	102	102	116	69	36	24	313	903	0.08
0.8%-0.9%		0	5	32	75	59	65	85	58	28	16	386	809	0.03
0.9%-1.0%		0	1	30	35	46	41	52	43	23	13	381	665	0.02
>1.0%		0	5	54	85	125	150	133	108	57	42	4505	5264	0.86
Total		1297	3938	2659	2052	1372	1048	853	481	232	141	6298	0.49	
Precision		0.73	0.53	0.37	0.33	0.23	0.24	0.21	0.14	0.12	0.09	0.72		

Table C.17. Confusion matrix of the KNN algorithm with CAV.

KNN – CAV		Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
Actual Class	0.0%-0.1%	812	547	102	21	4	0	0	0	0	0	0	1486	0.55
	0.1%-0.2%	342	1955	346	169	90	53	15	12	2	4	20	3008	0.65
	0.2%-0.3%	31	827	862	234	111	59	51	15	3	5	110	2308	0.37
	0.3%-0.4%	10	470	370	523	162	89	51	23	11	12	199	1920	0.27
	0.4%-0.5%	1	227	253	210	272	117	85	34	16	6	235	1456	0.19
	0.5%-0.6%	1	141	188	204	169	200	91	31	26	18	289	1358	0.15
	0.6%-0.7%	0	60	137	133	124	102	156	67	29	13	373	1194	0.13
	0.7%-0.8%	0	31	61	89	92	69	89	57	26	14	375	903	0.06
	0.8%-0.9%	0	8	37	80	57	45	70	31	30	13	438	809	0.04
	0.9%-1.0%	0	7	25	50	46	31	35	35	9	14	413	665	0.02
>1.0%	0	8	51	91	99	99	112	75	38	35	4656	5264	0.88	
Total		1197	4281	2432	1804	1226	864	755	380	190	134	7108	0.47	
Precision		0.68	0.46	0.35	0.29	0.22	0.23	0.21	0.15	0.16	0.10	0.66		

Table C.18. Confusion matrix of the NB algorithm with PGA.

NB – PGA	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0% 0.1%	0.1% 0.2%	0.2% 0.3%	0.3% 0.4%	0.4% 0.5%	0.5% 0.6%	0.6% 0.7%	0.7% 0.8%	0.8% 0.9%	0.9% 1.0%	>1.0%		
0.0%-0.1%		862	486	135	3	0	0	0	0	0	0	0	1486	0.58
0.1%-0.2%		363	1761	761	41	50	3	10	1	0	0	18	3008	0.59
0.2%-0.3%		16	1071	816	61	73	23	73	7	0	0	168	2308	0.35
0.3%-0.4%		0	643	481	122	166	54	60	18	0	0	376	1920	0.06
0.4%-0.5%		0	308	297	109	169	82	46	10	0	0	435	1456	0.12
0.5%-0.6%		0	215	192	93	175	94	74	21	0	0	494	1358	0.07
0.6%-0.7%		0	145	119	37	88	90	82	50	1	0	582	1194	0.07
0.7%-0.8%		0	66	66	26	84	63	53	38	1	0	506	903	0.04
0.8%-0.9%		0	30	47	10	47	42	44	30	3	0	556	809	0.00
0.9%-1.0%		0	22	20	5	27	31	29	13	0	0	518	665	0.00
>1.0%		0	56	51	12	133	132	129	44	7	0	4700	5264	0.89
Total		1241	4803	2985	519	1012	614	600	232	12	0	8353	0.42	
Precision		0.69	0.37	0.27	0.24	0.17	0.15	0.14	0.16	0.25		0.56		

Table C.19. Confusion matrix of the NB algorithm with PGV.

NB – PGV		Predicted Class											Total	Recall	
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	905	407	169	5	0	0	0	0	0	0	0	1486	0.61
		0.1%-0.2%	557	1645	677	96	14	0	3	0	0	0	16	3008	0.55
		0.2%-0.3%	42	909	786	122	124	32	23	1	0	0	269	2308	0.34
		0.3%-0.4%	27	628	510	120	64	53	35	4	0	0	479	1920	0.06
		0.4%-0.5%	2	346	330	126	61	42	46	14	0	0	489	1456	0.04
		0.5%-0.6%	0	250	276	110	101	64	67	15	0	0	475	1358	0.05
		0.6%-0.7%	0	114	197	81	109	76	74	6	0	0	537	1194	0.06
		0.7%-0.8%	0	52	123	45	80	79	50	18	0	0	456	903	0.02
		0.8%-0.9%	0	19	105	38	54	65	34	4	1	0	489	809	0.00
		0.9%-1.0%	0	10	53	12	30	54	18	5	0	0	483	665	0.00
>1.0%		0	10	104	27	77	264	104	19	3	0	4656	5264	0.88	
Total		1533	4390	3330	782	714	729	454	86	4	0	8349	0.41		
Precision		0.59	0.37	0.24	0.15	0.09	0.09	0.16	0.21	0.25		0.56			

Table C.20. Confusion matrix of the NB algorithm with PGD.

NB – PGD	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		809	506	147	2	1	0	0	0	0	0	21	1486	0.54
	0.0%-0.1%	809	506	147	2	1	0	0	0	0	0	21	1486	0.54
	0.1%-0.2%	456	1703	541	26	75	30	0	0	0	0	177	3008	0.57
	0.2%-0.3%	96	932	578	23	90	32	16	2	0	0	539	2308	0.25
	0.3%-0.4%	64	625	340	25	91	42	47	16	0	0	670	1920	0.01
	0.4%-0.5%	16	437	188	13	60	44	54	17	0	0	627	1456	0.04
	0.5%-0.6%	0	380	172	17	97	50	52	18	0	0	572	1358	0.04
	0.6%-0.7%	0	222	155	13	107	62	56	24	0	0	555	1194	0.05
	0.7%-0.8%	0	114	119	11	71	51	46	30	0	0	461	903	0.03
	0.8%-0.9%	0	92	100	4	29	42	21	20	0	0	501	809	0.00
	0.9%-1.0%	0	37	61	2	40	36	15	6	0	0	468	665	0.00
	> 1.0%	0	136	201	6	95	154	86	17	0	0	4569	5264	0.87
Total		1441	5184	2602	142	756	543	393	150	0	0	9160	0.39	
Precision		0.56	0.33	0.22	0.18	0.08	0.09	0.14	0.20			0.50		

Table C.21. Confusion matrix of the NB algorithm with $S_a(T_1)$.

NB - S _a (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%		
0.0%-0.1%		907	452	127	0	0	0	0	0	0	0	0	1486	0.61
0.1%-0.2%		367	1868	718	15	4	1	0	0	0	0	35	3008	0.62
0.2%-0.3%		15	870	924	110	52	28	27	3	0	0	279	2308	0.40
0.3%-0.4%		0	506	601	171	115	44	39	10	0	0	434	1920	0.09
0.4%-0.5%		0	274	309	129	119	58	51	15	0	0	501	1456	0.08
0.5%-0.6%		0	212	182	107	132	71	85	21	0	0	548	1358	0.05
0.6%-0.7%		0	137	111	65	76	73	94	47	0	0	591	1194	0.08
0.7%-0.8%		0	65	37	40	67	71	62	37	0	0	524	903	0.04
0.8%-0.9%		0	40	33	32	40	53	42	22	0	0	547	809	0.00
0.9%-1.0%		0	24	12	11	26	40	26	13	0	0	513	665	0.00
>1.0%		0	68	20	26	127	252	123	40	0	0	4608	5264	0.88
Total		1289	4516	3074	706	758	691	549	208	0	0	8580	0.43	
Precision		0.70	0.41	0.30	0.24	0.16	0.10	0.17	0.18			0.54		

Table C.22. Confusion matrix of the NB algorithm with $S_d(T_1)$.

NB - S _d (T ₁)	Actual Class	Predicted Class											Total	Recall	
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%			
0.0%-0.1%		1023	440	23	0	0	0	0	0	0	0	0	0	1486	0.69
0.1%-0.2%		365	2027	565	40	9	0	1	0	0	0	0	1	3008	0.67
0.2%-0.3%		16	779	960	392	83	6	12	5	0	0	0	55	2308	0.42
0.3%-0.4%		0	245	661	563	200	42	46	10	1	0	0	152	1920	0.29
0.4%-0.5%		0	86	311	372	229	66	85	26	1	0	0	280	1456	0.16
0.5%-0.6%		0	33	152	263	248	121	119	48	3	0	0	371	1358	0.09
0.6%-0.7%		0	8	92	149	149	124	109	80	12	0	0	471	1194	0.09
0.7%-0.8%		0	2	36	65	99	85	92	53	12	0	0	459	903	0.06
0.8%-0.9%		0	2	21	55	65	70	52	35	9	0	0	500	809	0.01
0.9%-1.0%		0	0	8	35	38	47	27	32	8	0	0	470	665	0.00
>1.0%		0	1	30	66	63	122	155	125	55	0	0	4647	5264	0.88
Total		1404	3623	2859	2000	1183	683	698	414	101	0	0	7406	0.48	
Precision		0.73	0.56	0.34	0.28	0.19	0.18	0.16	0.13	0.09			0.63		

Table C.23. Confusion matrix of the NB algorithm with CAV.

NB – CAV		Predicted Class												Total	Recall	
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%				
		0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%					
Actual Class		0.0%-0.1%	847	451	188	0	0	0	0	0	0	0	0	1486	0.57	0.43
		0.1%-0.2%	351	1900	627	30	24	6	8	0	0	0	62	3008	0.63	
		0.2%-0.3%	12	1038	769	68	104	24	43	5	0	0	245	2308	0.33	
		0.3%-0.4%	7	607	500	116	158	28	62	20	1	0	421	1920	0.06	
		0.4%-0.5%	0	309	355	97	143	55	57	27	0	0	413	1456	0.10	
		0.5%-0.6%	0	230	248	112	160	72	65	39	0	0	432	1358	0.05	
		0.6%-0.7%	0	112	162	64	108	70	66	81	1	0	530	1194	0.06	
		0.7%-0.8%	0	56	72	28	73	56	61	63	0	0	494	903	0.07	
		0.8%-0.9%	0	21	58	17	35	45	51	42	0	0	540	809	0.00	
Total		0.9%-1.0%	0	16	21	4	24	34	26	17	0	0	523	665	0.00	
		>1.0%	0	30	66	53	112	115	111	67	0	0	4710	5264	0.89	
		0.70	0.40	0.25	0.20	0.15	0.14	0.12	0.17	0.00	0	0.56				

Table C.24. Confusion matrix of the SGB algorithm with PGA.

SGB – PGA		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
0.0%-0.1%		1179	299	8	0	0	0	0	0	0	0	0	1486	0.79	
0.1%-0.2%		167	2495	317	14	4	9	9	2	0	0	0	3008	0.83	
0.2%-0.3%		3	627	1397	181	43	9	20	4	0	0	24	2308	0.61	
0.3%-0.4%		0	166	509	725	284	42	22	7	0	0	165	1920	0.38	
0.4%-0.5%		0	62	134	323	543	150	35	9	0	0	200	1456	0.37	
0.5%-0.6%		0	45	102	165	283	337	109	39	0	0	278	1358	0.25	
0.6%-0.7%		0	34	83	65	99	189	240	63	8	1	412	1194	0.20	
0.7%-0.8%		0	12	35	44	75	88	96	72	55	5	421	903	0.08	
0.8%-0.9%		0	10	24	29	30	37	52	29	63	16	519	809	0.08	
0.9%-1.0%		0	6	15	13	11	19	23	17	16	10	535	665	0.02	
>1.0%		0	9	19	33	31	36	32	23	25	8	5048	5264	0.96	
Total		1349	3765	2643	1592	1403	916	631	263	167	40	7602	0.59		
Precision		0.87	0.66	0.53	0.46	0.39	0.37	0.38	0.27	0.38	0.25	0.66			

Table C.25. Confusion matrix of the SGB algorithm with PGV.

SGB – PGV		Predicted Class												Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
0.0%-0.1%		1138	334	11	3	0	0	0	0	0	0	0	1486	0.77	
0.1%-0.2%		135	2449	360	29	27	8	0	0	0	0	0	3008	0.81	
0.2%-0.3%		12	632	1232	273	58	49	3	0	0	0	49	2308	0.53	
0.3%-0.4%		6	188	482	761	239	33	14	1	0	3	193	1920	0.40	
0.4%-0.5%		0	99	169	304	444	106	64	13	0	0	257	1456	0.30	
0.5%-0.6%		0	62	133	157	241	240	155	33	2	0	335	1358	0.18	
0.6%-0.7%		0	16	89	73	133	136	257	72	20	3	395	1194	0.22	
0.7%-0.8%		0	9	82	31	65	40	103	108	32	4	429	903	0.12	
0.8%-0.9%		0	4	51	11	27	13	54	65	54	14	516	809	0.07	
0.9%-1.0%		0	1	25	9	11	7	27	21	14	32	518	665	0.05	
>1.0%		0	2	64	21	22	11	64	33	11	20	5016	5264	0.95	
Total		1291	3796	2698	1672	1267	643	741	346	133	76	7708	0.58		
Precision		0.88	0.65	0.46	0.46	0.35	0.37	0.35	0.31	0.41	0.42	0.65			

Table C.26. Confusion matrix of the SGB algorithm with PGD.

SGB – PGD	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
0.0%-0.1%	1118	346	13	5	4	0	0	0	0	0	0	1486	0.75	
0.1%-0.2%	188	2473	283	36	17	0	4	0	0	0	7	3008	0.82	
0.2%-0.3%	34	616	1110	247	57	17	16	0	0	0	211	2308	0.48	
0.3%-0.4%	12	288	375	710	154	27	24	0	0	0	330	1920	0.37	
0.4%-0.5%	1	193	165	226	357	100	47	6	0	0	361	1456	0.25	
0.5%-0.6%	4	171	135	101	195	219	67	17	7	0	442	1358	0.16	
0.6%-0.7%	10	91	124	78	94	85	164	38	27	4	479	1194	0.14	
0.7%-0.8%	7	43	91	42	40	19	72	55	48	6	480	903	0.06	
0.8%-0.9%	2	52	70	34	11	12	12	29	32	11	544	809	0.04	
0.9%-1.0%	0	26	36	17	12	6	15	11	15	11	516	665	0.02	
>1.0%	5	100	99	63	9	41	40	18	12	7	4870	5264	0.93	
Total	1381	4399	2501	1559	950	526	461	174	141	39	8240	0.55		
Precision	0.81	0.56	0.44	0.46	0.38	0.42	0.36	0.32	0.23	0.28	0.59			

Table C.27. Confusion matrix of the SGB algorithm with $S_a(T_1)$.

SGB – $S_a(T_1)$	Actual Class	Predicted Class												Total	Recall
		0.0% -	0.1% -	0.2% -	0.3% -	0.4% -	0.5% -	0.6% -	0.7% -	0.8% -	0.9% -	1.0% -	>1.0%		
		0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	1.0%	>1.0%		
0.0%-0.1%		1019	455	12	0	0	0	0	0	0	0	0	0	1486	0.69
0.1%-0.2%		258	2302	392	26	23	2	2	0	0	0	0	3	3008	0.77
0.2%-0.3%		6	804	1063	284	83	10	23	2	1	0	0	32	2308	0.46
0.3%-0.4%		0	260	614	680	174	50	30	13	0	0	0	99	1920	0.35
0.4%-0.5%		0	84	334	394	261	82	78	9	4	1	0	209	1456	0.18
0.5%-0.6%		0	33	170	285	225	173	100	17	1	0	0	354	1358	0.13
0.6%-0.7%		0	12	131	163	89	115	137	34	3	1	0	509	1194	0.11
0.7%-0.8%		0	4	42	96	70	60	72	32	4	0	0	523	903	0.04
0.8%-0.9%		0	4	27	74	60	40	27	19	2	0	0	556	809	0.00
0.9%-1.0%		0	0	9	41	39	18	19	12	2	0	0	525	665	0.00
>1.0%		0	0	31	78	52	46	71	15	0	2	0	4969	5264	0.94
Total		1283	3958	2825	2121	1076	596	559	153	17	4	0	7779	0.52	
Precision		0.79	0.58	0.38	0.32	0.24	0.29	0.25	0.21	0.12	0.00	0.00	0.64		

Table C.28. Confusion matrix of the SGB algorithm with $S_d(T_1)$.

SGB – S _d (T ₁)	Actual Class	Predicted Class											Total	Recall
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%		
		0.0%	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%		
0.0%-0.1%		1031	443	12	0	0	0	0	0	0	0	0	1486	0.69
0.1%-0.2%		227	2256	444	58	18	4	1	0	0	0	0	3008	0.75
0.2%-0.3%		1	656	1102	366	133	19	16	0	0	1	14	2308	0.48
0.3%-0.4%		0	169	550	744	269	81	29	8	4	5	61	1920	0.39
0.4%-0.5%		0	41	252	416	372	143	58	10	2	1	161	1456	0.26
0.5%-0.6%		1	9	103	282	337	228	76	17	4	1	300	1358	0.17
0.6%-0.7%		0	3	82	129	211	166	114	36	5	1	447	1194	0.10
0.7%-0.8%		0	1	14	76	134	79	82	26	5	0	486	903	0.03
0.8%-0.9%		0	2	15	53	103	62	28	17	3	0	526	809	0.00
0.9%-1.0%		0	0	5	24	69	44	25	11	2	0	485	665	0.00
>1.0%		0	0	18	38	116	120	73	11	4	1	4883	5264	0.93
Total		1260	3580	2597	2186	1762	946	502	136	29	10	7363	0.53	
Precision		0.82	0.63	0.42	0.34	0.21	0.24	0.23	0.19	0.10	0.00	0.66		

Table C.29. Confusion matrix of the SGB algorithm with CAV.

SGB – CAV		Predicted Class											Total	Recall	
		0.0% - 0.1%	0.1% - 0.2%	0.2% - 0.3%	0.3% - 0.4%	0.4% - 0.5%	0.5% - 0.6%	0.6% - 0.7%	0.7% - 0.8%	0.8% - 0.9%	0.9% - 1.0%	>1.0%			
Actual Class		0.0%-0.1%	1124	335	25	0	2	0	0	0	0	0	0	1486	0.76
		0.1%-0.2%	143	2534	244	24	17	4	16	2	0	0	24	3008	0.84
		0.2%-0.3%	5	667	1206	263	30	28	12	16	0	0	81	2308	0.52
		0.3%-0.4%	0	222	439	759	278	37	12	15	0	0	158	1920	0.40
		0.4%-0.5%	0	113	161	298	525	106	49	8	3	0	193	1456	0.36
		0.5%-0.6%	0	59	101	162	278	296	132	21	5	1	303	1358	0.22
		0.6%-0.7%	0	26	75	63	99	190	229	86	14	2	410	1194	0.19
		0.7%-0.8%	0	6	44	39	44	87	96	89	35	9	454	903	0.10
		0.8%-0.9%	0	9	34	35	23	29	30	42	55	37	515	809	0.07
		0.9%-1.0%	0	5	10	24	11	11	13	11	25	41	514	665	0.06
>1.0%		0	6	28	73	35	32	81	17	22	17	4953	5264	0.94	
Total		1272	3982	2367	1740	1342	820	670	307	159	107	7605	0.58		
Precision		0.88	0.64	0.51	0.44	0.39	0.36	0.34	0.29	0.35	0.38	0.65			